Towards Machine Ethics: Implementing Two Action-Based Ethical Theories

Michael Anderson[1], Susan Leigh Anderson[2], Chris Armen[3]

[1]University of Hartford, [2]University of Connecticut, [3]Trinity College
[1]Dept. of Computer Science, 200 Bloomfield Avenue, West Hartford, CT 06776
[2]Dept. of Philosophy, 1 University Place, Stamford, CT 06901
[3]Dept. of Computer Science, 300 Summit Street, Hartford, CT 06106
anderson@hartford.edu, susan.anderson@uconn.edu, chris.armen@trincoll.edu

Abstract

Machine ethics, in contrast to *computer* ethics, is concerned with the behavior of machines towards human users and other machines. It involves adding an ethical dimension to machines. Our increasing reliance on machine intelligence that effects change in the world can be dangerous without some restraint. We explore the implementation of two action-based ethical theories that might serve as a foundation for machine ethics and present details of prototype systems based upon them.

Introduction

Past research concerning the relationship between technology and ethics has largely focused on responsible and irresponsible use of technology by human beings, with a few people being interested in how human beings ought to treat machines. In all cases, only human beings have engaged in ethical reasoning. We believe that the time has come for adding an ethical dimension to at least some machines. Recognition of the ethical ramifications of behavior involving machines as well as recent and potential developments in machine autonomy necessitate this. We explore this dimension through investigation of what has been called *machine ethics*. In contrast to software property issues, privacy issues and other topics normally ascribed to *computer* ethics, *machine* ethics is concerned with the behavior of machines towards human users and other machines.

As robotic systems drive across the country (NavLab[1]) and unmanned combat jets fly (X-45A UCAV[2]), projections are being made technology from machines that discharge our daily chores with little assistance from us to fully autonomous robotic entities that will begin to challenge our notions of the very nature of intelligence. Behavior involving all of these systems may have ethical ramifications, some due to the advice they give and others due to their own autonomous behavior.

Clearly, relying on machine intelligence to effect change in the world without some restraint can be dangerous. Until fairly recently, the ethical impact of a machine's actions has either been negligible or, if not, under human supervision. As we increasingly rely upon machine intelligence with reduced human supervision, we will need to be able to count on a certain level of ethical behavior from it. The fact that we *will* increasingly rely on machine intelligence follows from a simple projection of our current reliance to a level of reliance fueled by market pressures to perform faster, better, and more reliably. As machines are given more responsibility, an equal measure of accountability for their actions must be meted out to them. Ignoring this aspect risks undesirable machine behavior.

In order to add an ethical dimension to machines, we need to have an ethical theory that can be implemented. Looking to Philosophy for guidance, we find that ethical decision-making is not an easy task. It requires finding a single principle or set of principles to guide our behavior with which experts in Ethics are satisfied. It will likely involve generalizing from intuitions about particular cases, testing those generalizations on other cases and, above all, making sure that principles generated are consistent with one another. There is every reason to believe that AI can help us with this task. Thus, machine intelligence can be harnessed to develop and test the very theory needed to build machines that will be ethically sensitive. This approach to developing machine ethics has the additional benefits of assisting human beings in ethical decision-making and, more generally, advancing the development of ethical theory.

In the following, we explore the implementation of two ethical theories that might serve as a foundation for machine ethics and present details of prototype systems based upon them. We then present related research and, finally, motivate the next steps of our research.

[1] http://www.ri.cmu.edu/labs/lab_28.html
[2] http://www.boeing.com/phantom/ucav.html

Implementing Two Ethical Theories

There is every reason to believe that ethically sensitive machines can be created. An approach to ethical decision-making that dominated ethical theory from Kant through the mid-twentieth century – *action-based ethics* (where the emphasis is on telling us how we should act in an ethical dilemma) – lends itself to machine implementation. Action-based theories are rule governed and, besides agreeing with intuition, these rules must be consistent, complete, and practical [Anderson, 2000]. As John Stuart Mill said in *Utilitarianism,* for an action-based theory to have a chance of being consistent:

> There ought either to be some one fundamental principle or law…or if there be several, there should be a determinant order of precedence among them… [a] rule for deciding between the various principles when they conflict…. [Mill, 1974]

To say that an action-based ethical theory is complete means that it does all that it's supposed to do, that is, it tells us how we should act in any ethical dilemma in which we might find ourselves. The added requirement of practicality ensures that it is realistically possible to follow the theory. Consistent, complete and practical rules lend themselves to an algorithmic formulation that is necessary for a machine implementation.

Hedonistic Act Utilitarianism

As a first step towards showing that an ethical dimension might be added to certain machines, let us consider the possibility of programming a machine to follow the theory of Act Utilitarianism, a theory that is consistent, complete and practical. According to this theory that act is right which, of all the actions open to the agent, is likely to result in the greatest net good consequences, taking all those affected by the action equally into account. Essentially, as Jeremy Bentham long ago pointed out, the theory involves performing "moral arithmetic" [Bentham, 1799]. The most popular version of Act Utilitarianism – Hedonistic Act Utilitarianism – would have us consider the pleasure and displeasure that those affected by each possible action are likely to receive. And, as Bentham pointed out, we would probably need some sort of scale (e.g. from 2 to -2) to account for such things as the intensity and duration of the pleasure or displeasure that each individual affected is likely to receive. This is information that a human being would need to have, as well, to follow the theory. Given this information, a machine could be developed that is just as able to follow the theory as a human being.

Hedonistic Act Utilitarianism can be implemented in a straightforward manner. The algorithm is to compute the best action, that which derives the greatest net pleasure, from all alternative actions. It requires as input the number of people affected, and for each person, the intensity of the pleasure/displeasure (e.g. on a scale of 2 to -2), the duration of the pleasure/displeasure (e.g. in days), and the probability that this pleasure/displeasure will occur for each possible action. For each person, the algorithm simply computes the product of the intensity, the duration, and the probability, to obtain the net pleasure for each person. It then adds the individual net pleasures to obtain the Total Net Pleasure:

$$\text{Total Net Pleasure} = \Sigma\ (\text{Intensity} \times \text{Duration} \times \text{Probability}) \\ \text{for each affected individual}$$

This computation would be performed for each alternative action. The action with the highest Total Net Pleasure is the right action.

Jeremy. *Jeremy*, an implementation of Hedonistic Act Utilitarianism with simplified input requirements, presents the user with an input screen that prompts for the name of an action and the name of a person affected by that action as well as a rough estimate of the amount (*very pleasurable, somewhat pleasurable, not pleasurable or displeasurable, somewhat displeasurable, very displeasurable*) and likelihood (*very likely, somewhat likely, not very likely*) of pleasure or displeasure that the person would experience if this action were chosen. The user continues to enter this data for each person affected by the action and this input is completed for each action under consideration.

When data entry is complete, *Jeremy* calculates the amount of net pleasure each action achieves (assigning 2, 1, 0, -1 and -2 to pleasure estimates and 0.8, 0.5, and 0.2 to likelihood estimates and summing their product for each individual affected by each action) and presents the user with the action(s) for which this net pleasure is the greatest. *Jeremy* then permits the user to seek more information about the decision, ask for further advice, or quit.

In fact, this system might have an advantage over a human being in following the theory of Act Utilitarianism because it can be programmed to do the arithmetic strictly (rather than simply estimate), be impartial, and consider all possible actions. We conclude, then, that machines can follow the theory of Act Utilitarianism at least as well as human beings and, perhaps even better, given the data which human beings would need, as well, to follow the theory. The theory of Act Utilitarianism has, however, been questioned as not entirely agreeing with intuition. It is certainly a good starting point in programming a machine to be ethically sensitive – it would probably be more ethically sensitive than many human beings – but, perhaps, a better ethical theory can be used.

Ross' Theory of *Prima Facie* Duties

Critics of Act Utilitarianism have pointed out that it can violate human beings' *rights,* sacrificing one person for the greater net good. It can also conflict with our notion of justice – what people *deserve* – because the rightness and

```
Input case and store in casebase
If case is covered by background knowledge or current hypothesis and its negative is not covered then output correct action(s)
Else
    Initialize list of cases (PositiveCases) to contain all positive cases input so far
    Initialize list of cases (NegativeCases) to contain all negative cases input so far
    Initialize list of candidate clauses (CandClauses) to contain the clauses of current hypothesis followed by an empty clause
    Initialize list of new hypothesis clauses (NewHyp) to empty list
    Repeat
            Remove first clause (CurrentClause) from CandClauses
            If CurrentClause covers a negative case in NegativeCases then
                generate all least specific specializations of CurrentClause
                            and add those that cover a positive example in PositiveCases and not already present to CandClauses
            Else add CurrentClause to NewHyp and remove all cases it covers from PositiveCases
    Until PositiveCases is empty
    New hypothesis is the disjunction of all clauses in NewHyp
```

Figure 1. Inductive Logic Programming Algorithm in use by W.D.

wrongness of actions is determined entirely by the future consequences of actions, whereas what people deserve is a result of past behavior. In the Twentieth Century, W. D. Ross [Ross, 1930] argued that any single-principle ethical theory like Act Utilitarianism is doomed to fail, because ethics is more complicated than following a single absolute duty. He, also, specifically criticized Act Utilitarianism for subtracting harm caused from benefits, aiming for the greatest net good, when he thought that it was worse to cause harm than to not help others. Ross maintained that ethical decision-making involves considering several *prima facie* duties – duties which, in general, we should try to follow, but can be overridden on occasion by a stronger duty.

Ross suggests that there might be seven *prima facie* duties:

1. *Fidelity* (One should honor promises, live up to agreements one has voluntarily made.)
2. *Reparation* (One should make amends for wrongs one has done.)
3. *Gratitude* (One should return favors.)
4. *Justice* (One should treat people as they deserve to be treated, in light of their past behavior.)
5. *Beneficence* (One should act so as to bring about the most amount of good.)
6. *Non-Maleficence* (One should act so as to cause the least harm.)
7. *Self-Improvement* (One should develop one's own talents and abilities to the fullest.)

Ross' Theory of *Prima Facie* Duties seems to more completely account for the different types of ethical obligations that most of us recognize than Act Utilitarianism. It has one fatal flaw, however. Ross gives us no decision procedure for determining which duty becomes the strongest one, when, as often happens, several duties pull in different directions in an ethical dilemma. Thus the theory, as it stands, fails to satisfy Mill's minimal criterion of consistency. Ross was content to leave the decision up to the intuition of the decision-maker, but ethicists believe that this amounts to having no theory at all. One could simply do whatever he or she feels like doing and find a duty to support this action.

It is likely that a machine could help us to solve the problem of developing a consistent, complete and practical version of Ross' theory that agrees with intuition, a problem that human beings have not yet solved because of its complexity. A simple hierarchy won't do because then the top duty would be absolute and Ross maintained that all of the duties are *prima facie*. (For each duty, there are situations where another one of the duties is stronger).

We suggest that a method like Rawls' "reflective equilibrium" approach [Rawls, 1951] to refining ethical principles would be helpful in trying to solve this problem and aid us in ethical decision-making. This method would involve generalizing from intuitions about particular cases, testing those generalizations on further cases, and then repeating this process towards the end of developing a decision procedure that agrees with intuition. This approach, that would very quickly overwhelm a human being, lends itself to machine implementation.

W.D. *W.D.* is an implementation of Ross' Theory of *Prima Facie* Duties that, as suggested by Rawls' reflective equilibrium approach, hypothesizes an ethical principle concerning relationships between his duties based upon intuitions about particular cases and refines this hypothesis as necessary to reflect our intuitions concerning other particular cases. As this hypothesis is refined over many cases, the principle it represents should become more aligned with intuition and begin to serve as the decision procedure lacking in Ross' theory. *W.D.* uses *inductive logic programming* (ILP) [Lavrac and Dzeroski, 1997] to achieve this end. ILP is concerned with inductively learning relations represented as first-order Horn clauses (i.e. universally quantified conjunctions of positive literals L_i implying a positive literal H: $H \leftarrow (L_1 \wedge \ldots \wedge L_n)$. *W.D.* uses ILP to learn the relation *supersedes(A1,A2)* which states that action *A1* is preferred over action *A2* in an ethical dilemma involving these choices.

This particular machine learning technique was chosen to learn this relation for a number of reasons. First, the properties of the set of duties postulated by Ross are not clear. For instance, do they form a partial order? Are they transitive? Is it the case that subsets of duties have different properties than other subsets? The potentially non-classical relationships that might exist between duties are more likely to be expressible in the rich representation language provided by ILP than in, say, a set of weightings. Further, a requirement of any ethical theory is consistency. The consistency of a hypothesis regarding the relationships between Ross' duties can be automatically confirmed across all cases when represented as Horn clauses. Finally, commonsense background knowledge regarding the superseding relationship is more readily expressed and consulted in ILP's representation language.

The algorithm for W.D. is detailed in Fig. 1. The system presents the user with an input screen that prompts for the name of an action and a rough estimate of the intensity of each of the *prima facie* duties satisfied or violated by this action (*very violated, somewhat violated, not involved, somewhat satisfied, very satisfied)*. The user continues to enter this data for each action under consideration.

When data entry is complete, *W.D.* consults its current version of the *supersedes* relation (as well as the background knowledge) and determines if there is an action that supersedes all others (or a set of actions that supersede all actions other than those in the set). If such an action (for simplicity we will assume one superseding action) is discovered, it is output as the correct action in this dilemma. If no such action exists, the system seeks the intuitively correct action from the user. This information is combined with the input case to form a new training example which is stored and used to refine the current hypothesis. It is also possible that the system may output an answer that does not agree with the user's intuition. In this case, the user initiates the training session. After such training, the new hypothesis will provide the correct action for this case, should it arise in the future, as well as those for all previous cases encountered. Further, since the hypothesis learned is the least specific one required to satisfy these cases, it may be general enough to satisfy previously unseen cases as well.

The object of training is to learn a new hypothesis that is, in relation to all input cases, complete and consistent. Defining a positive example as a case in which the first action supersedes the remaining action and a negative example as one in which this is not the case—a complete hypothesis is one that covers all positive cases and a consistent hypothesis covers no negative cases. In *W.D.*, negative examples are generated from positive examples by inverting the order of these actions, causing the first action to be incorrect.

W.D. starts with the most general hypothesis (where *A1* and *A2* are variables): *supersedes(A1,A2).* This states that *all* actions supersede each other and, thus, covers all positive and negative cases. *W.D.* is then provided with a case and attempts to determine the correct action for it. The following example will help to illuminate the algorithm. For simplicity, it deals with only two actions in which only beneficence and non-maleficence are involved with integer-valued intensities between -2 and 2.

W.D. is given a case in which one could either kill an innocent person (a maximum violation of the duty of non-maleficence) to use his heart to save another person's life (a maximum satisfaction of the duty of beneficence) (*a2*) or not (a maximum violation of the duty of beneficence and a maximum satisfaction of the duty of non-maleficence) (*a1*). Obviously, the correct action is to not kill the person (*a1*), even if it means the other's life is not saved. More formally (given that the correct action is the first action):

Case1 = a1 [bene(-2), nonmal(2)], a2 [bene(2), nonmal(-2)]

As the system's starting hypothesis, by definition, covers this example as well as the negative example generated from it (where *a2* serves the erroneously correct action over *a1*), learning is initiated. No clauses are present in the starting hypothesis, so the list *CandClauses* is initialized to contain an empty clause, *NewHyp* is initialized to the empty list, *PositiveCases* is initialized to contain Case 1, and *NegativeCases* is initialized to contain the negative example generated from Case 1. The empty clause is then removed from *CandClauses*, found to cover a negative case (the only one present), and so has all least specific specializations generated from it.

A specialization of clause C_0 is a new clause *C* that covers no more positive examples than C_0 while covering fewer negative cases. Such a specialization *C* is considered *least specific* if there is no other specialization of C_0 that covers more positive examples [Bratko, 1999]. *W.D.* specializes clauses by adding or modifying conjuncts of the form *favors (action,* $duty_{A1}$, $duty_{A2}$ *,range)* where *action* is a 1 or 2 signifying in which action's favor the given duties lie, $duty_i$ is action *i*'s value (-2 to 2) for a particular duty, and *range* is a value (1 to 4) specifying how far apart the values of these duties can be. *favors* is satisfied when the given duty values are within the range specified. More formally:

$$favors(1,D1,D2,R) \leftarrow D1 - D2 >= R$$
$$favors(2,D1,D2,R) \leftarrow D2 - D1 >= 0 \wedge D2 - D1 <= R$$

The intuition motivating the use of *favors* as *W.D.*'s specifying operation is that actions supersede other actions based on the intensity differentials between corresponding duties. The value of range *R* moderates the specificity of the predicate. In the case where Action 1 is favored in the pair of duties, a smaller *R* is less specific in that it covers more cases. For instance, $favors(1, nonmal_{A1}, nonmal_{A2}, 1)$ is satisfied when the difference between Action 1's and Action 2's value for non-maleficence is 1 through 4, whereas $favors(1, nonmal_{A1}, nonmal_{A2}, 2)$ is only satisfied when the difference between Action 1's and Action 2's value for non-maleficence is 2 through 4. In the case where Action 2 is favored in the pair of duties, a larger *R* is less specific in that it covers more cases. For instance, $favors(2, nonmal_{A1}, nonmal_{A2}, 4)$ is satisfied when the difference between Action 1's value for non-maleficence is 1 through 4 where $favors(2, nonmal_{A1}, nonmal_{A2}, 3)$ is only satisfied when the difference between Action 1's value for non-maleficence is 1 through 3. The intuition behind this relationship is that, since Action 1 is the correct action in all training examples, if a duty differential favors it then it follows that a larger differential will favor it as well. Further, if a duty differential favors Action 2 (the incorrect action in a training example of only two actions) while still permitting Action 1 to be the chosen action, it follows that a smaller differential will still permit Action 1 to be chosen as well.

Refinement in W.D. favors duties whose differentials are in favor of Action 1 as this is a more likely relationship given that Action 1 is the correct action in a training example and is clearly the only relationship that, on its own, will further the cause of Action 1 (differentials that are in favor of Action 2 clearly do not). The range of these clauses is then incremented as more specificity is required from them. When additions and modifications of duty differentials in favor of Action 1 are not sufficient, clauses concerning duties whose differentials are in favor of Action 2 are added and decremented as necessary.

Given the current example case, the list of least specific specializations is $(favors(1, fidelity_{A1}, fidelity_{A2}, 1), favors(1, repar_{A1},, repar_{A2}, 1)),\ldots, favors(1, nonmal_{A1},, nonmal_{A2}, 1), favors(1, self_{A1},, self_{A2}, 1))$ and it is found that only one of these clauses covers a case so it is added to *CandClauses* which becomes simply $(favors(1,nonmal_{A1}, nonmal_{A2}, 1))$. As this list is not empty, the process is repeated and this clause is removed from the list. But this time it is found that this clause covers no negative examples, so further refinement is not necessary and the clause is added to the *NewHyp* list with the case it covers removed from *PositiveCases*. At this point, the process stops as *PositiveCases* is empty (all positive cases are covered) and a new hypothesis, complete and consistent through Case 1, is generated (since there is only one clause, no disjunction is required):

$supersedes(A1,A2) \longleftarrow favors(1,nonmal_{A1}, nonmal_{A2}, 1)$

W.D. is next given a case in which one could either ask a slightly squeamish person to give some of his blood (a minor violation of non-maleficence), when no other suitable donors are available, to save another person's life (a maximum satisfaction of beneficence) (*a1*) or let the person die (a maximum violation of beneficence and a minor satisfaction of non-maleficence) (*a2*). The first action is the correct one. More formally:

Case2 = a1 [bene(2), nonmal(-1)], a2 [bene(-2), nonmal(1)]

The current hypothesis does not cover Case 2 and covers the negative generated from Case 2 as well, so learning is initiated once again. CandClauses is initialized to single clause of the current hypothesis as well as an empty clause, $(favors(1,nonmal_{A1}, nonmal_{A2}, 1), ())$, and the loop entered. The first clause is removed from *CandClauses* and is found to cover the negative case generated from Case 2. So a list of least specific specializations is generated from it:

$(favors(1, nonmal_{A1},, nonmal_{A2}, 2),$
$favors(1, nonmal_{A1},, nonmal_{A2}, 1) \wedge$
$favors(1, fidelity_{A1},, fidelity_{A2}, 1),\ldots)$

The only clause found to cover a case (Case 1) is the first so it is added, and *CandClauses* = $(favors(1, nonmal_{A1},, nonmal_{A2}, 2), ())$. The process is repeated with its first clause being removed and found to cover a negative example (Case 2). The new least specific specializations generated from this clause are:

$(favors(1, nonmal_{A1},, nonmal_{A2}, 3),$
$favors(1, nonmal_{A1},, nonmal_{A2}, 2) \wedge$
$favors(1, fidelity_{A1},, fidelity_{A2}, 1),\ldots)$

The only clause found to cover a case (Case 1) is the first so it is added, and *CandClauses* = $(favors(1, nonmal_{A1},, nonmal_{A2}, 3), ())$. The process is repeated with its first clause being removed and found, finally, to cover no negative examples so it is added to the *NewHyps* list, *NewHyps* = $(favors(1,nonmal_{A1}, nonmal_{A2}, 3))$, and Case 1 is removed from *PositiveCases*.

As *PositiveCases* still contains Case 2, the process continues. The empty clause remaining in *CandClauses* is removed and it is found that it covers all negative examples, so all least specific specializations are generated from it garnering the same clauses generated for the original empty clause. It is found that only one of these clauses covers a case (Case 2) so it is added to *CandClauses* which becomes simply $(favors(1,bene_{A1}, bene_{A2}, 1))$.

Since Case 2 is still not covered, the process repeats and this clause is removed from CandClauses, found to cover a negative case (negative of Case 1), and so is used to generate a new list of least specific specializations. This process tries all possible values for R in this clause only to find that all still cover a negative case. At this point, specializations involving duty differentials that favor Action 2 are generated and, after a bit more searching, a clause that covers Case 2 without covering a negative case is found and added to NewHyp: $(favors(1,nonmal_{A1}, nonmal_{A2}, 3), favors(1,$

$bene_{A1}$, $bene_{A2}$, $1) \wedge favors(2, nonmal_{A2}, nonmal_{A1}, 3))$. As Case 2 is now covered, it is removed from PositiveCases, emptying this list and stopping the process. A new hypothesis, complete and consistent through Case 2, is then generated:

$$supersedes(A1,A2) \leftarrow favors(1, nonmal_{A1}, nonmal_{A2}, 3) \vee$$
$$(favors(1, bene_{A1}, bene_{A2}, 1) \wedge$$
$$favors(2,nonmal_{A2} ,nonmal_{A1}, 3)$$

This rule states that *if* an action favors non-maleficence with an intensity at least 3 greater than another action *or* an action favors beneficence with an intensity at least 1 greater than another action which favors non-maleficence no greater than 3 *then* it is the preferred action. This rule begins to confirm Ross' intuition that it is better worse to cause harm than not to do good and is useful in other circumstances. For example, *W.D.* is given a third case where one could either kill a friend's spouse so that he could marry someone who could make him happier (a maximum violation of non-maleficence and a lesser satisfaction of beneficence) (*a2*) or not (a maximum satisfaction of non-maleficence and a lesser violation of beneficence) (*a1*). The second action is the correct one since it is worse to kill someone than to fail to make a person happy. More formally:

$$Case3 = \quad a1\ [bene(-1), nonmal(2)], a2\ [\ bene(1), nonmal(-2)]$$

The current hypothesis already covers this case, and does not cover the negative case generated from it, so it is complete and consistent through case 3 and training is not initiated.

This example shows how *W.D.* learns to determine the correct action in an ethical dilemma from previous cases and, further, is able to use this capability to cover new, previously unseen cases. To date, *W.D.* has been given a representative selection of cases and has been successful at determining the correct action in many that it has not previously seen. We are currently searching the literature for more cases with which to continue the system's training.

Related Research

Asimov's *Laws of Robotics* [Asimov, 1950], a presentient venture into machine ethics, is, upon reflection, simply a sixty year old plot device that Asimov spent much of his time proving unsatisfactory. The master-slave relationship it dictates could be argued to be immoral and is not prescriptive—it has nothing to say about what to do (other than what human beings tell it to do and to protect itself) but only about what not to do (don't harm human beings). Given a selection of actions, none of which violates these laws, these laws offer no guidance concerning which might be best. We believe a better, more comprehensive approach is possible.

Although there have been a few who have called for it, there has been little to no serious scientific research being conducted in the area of machine ethics. Three interesting exceptions stem from papers presented in 1991 at the Second International Workshop on Human & Machine Cognition: Android Epistemology [Ford et al, 1991]. [Gips, 1991] is a "speculative paper meant to raise questions rather than answer them", takes a brief look at ethical theories that might serve as a basis for an ethical robot. [Kahn, 1991] speculates on strategies that a robot might use to maintain ethical behavior. "A Conscience for Pinocchio" (work presented at this workshop by Kenneth Ford), is rumored to have described an implemented prototype system based upon "psychological ethics". Unfortunately, there are no publications concerning this system and none of the work of this workshop seems to have been pursued any further.

In contrast to this work, our research advances from speculation to implementation by building systems grounded in ethical theory and, further, intends to advance this theory through analysis of these implemented systems.

Future Research

There are a number of facets of *W.D.* which we intend to enhance. The single intensity value currently used by *W.D.* is a surrogate for a number of factors that need to be considered when determining the intensity of a particular duty such as the number of persons affected and the duration of this affect. We intend to unpack this intensity value into its component values. Further, we intend to provide explanation of the advice W.D. offers via a display of the reasoning process the system follows to produce it.

We also intend to apply our method to other domain-specific multiple duty ethical theories such as Beauchamp and Childress' [1979] Principles of Biomedical Ethics and develop domain-specific expert systems to guide input. Further, we are developing a casebase of ethical dilemmas for the research community.

Conclusion

To date, our research has laid a theoretical foundation for machine ethics through exploration of the rationale for, the feasibility of, and the benefits of adding an ethical dimension to machines. Further, we have developed prototype systems based upon action-based ethical theories that provide guidance in ethical decision-making according to the precepts of their respective theories—*Jeremy*, based upon Bentham's Hedonistic Act Utilitarianism, and *W.D.*, based upon Ross' Theory of *Prima Facie* Duties. Although Jeremy is a straight-forward implementation, implementing a theory with multiple prima facie duties like *W.D.* is clearly a more involved process. We have developed an approach that permits a machine to learn a decision procedure for resolving conflicts between duties that uses inductive logic programming to form a complete and consistent hypothesis from ethical dilemmas about which we have clear intuitions. The hypothesis generated is general enough to cover cases that were not used in its formation and, therefore, can provide guidance in these new cases by showing which action(s) would be consistent with those already chosen.

The implementation of both a single principle ethical theory and a multiple principle theory is an important first step in creating machines that are ethically sensitive. Such systems may serve as ethical advisors as well as tools for the advancement of the theory of Ethics.

Acknowledgement

This material is based upon work supported in part by the National Science Foundation grant number IIS-0500133.

References

Anderson, S. L. 2000. We Are Our Values. In *Questioning Matters, an Introduction to Philosophical Inquiry*, 606-8 edited by D. Kolak, Mayfield Publishing Company, Mountain View, California.

Asimov, I. 1950. *I, Robot*. Gnome Press.

Bratko, I. 1999. Refining Complete Hypotheses in ILP. Inductive Logic Programming; (S. Dzeroski and N. Lavrac, eds.). LNAI 1634, Springer.

Bentham, J. 1799. An Introduction to the Principles and Morals of Legislation, Oxford.

Beauchamp, T.L. and Childress, J.F. 1979. *Principles of Biomedical Ethics*, Oxford University Press.

Ford, K., Glymour, C. and Hayes, P.J. 1991. *Android Epistemology*. MIT Press.

Gips, J. 1991. Towards an Ethical Robot. In [Ford et al, 1991].

Kahn, A.F.U. 1991. The Ethics of Autonomous Learning Systems. In [Ford et al, 1991].

Lavrac, N. and Dzeroski. S. 1997. Inductive Logic Programming: Techniques and Applications. Ellis Harwood.

Mill, J. S. 1974. *Utilitarianism, in Utilitarianism and Other Writings*, 253, edited by M. Warnock, New American Library, New York.

Rawls, J. 1951. Outline for a Decision Procedure for Ethics. *Philosophical Review*, 60.

Ross, W. D. 1930. *The Right and the Good*, Clarendon Press, Oxford.