Formale Verifikation ethischer Entscheidungen bei autonomen Systemen

Benjamin Fresz
Institut für Automatisierungstechnik und Softwaresysteme (IAS)
Universität Stuttgart
Stuttgart, Deutschland
st148254@stud.uni-stuttgart.de

***Abstract*—Autonom agierende Agenten handeln zunehmend in räumlicher Nähe zu Menschen, wodurch das sichere und ethisch korrekte Verhalten der Agenten immer wichtiger wird. Um dieses Verhalten erzeugen und verifizieren zu können, wird ein Implementierungsansatz für Belief-Desire-Intention-Agenten vorgestellt und diskutiert. Der Ansatz basiert auf einem gegebenen ethischen Regelwerk, dem das Verhalten des Agenten zu folgen hat. Dies kann über eine Methode der formalen Verifikation - Model Checking - überprüft und garantiert werden. Um den vorgestellten Ansatz zu veranschaulichen, wird das ethische Verhalten einer unbemannten Drohne in zwei verschiedenen Situationen vorgestellt. Weiterhin werden Erweiterungsmöglichkeiten der formalen Verifikation auf flexiblere Reinforcement-Learning-Agenten aufgezeigt. Alle Punkte werden zusätzlich aus der Sicht möglicher Stakeholder betrachtet, für die der entstehende Wert der vorgestellten Konzepte hervorgehoben wird.**

***Index Terms*—formale Verifikation, Model Checking, autonome Systeme, Belief-Desire-Intention, Reinforcement Learning, Agent, Ethik, Dilemma, Trolley-Problem**

I. Einführung

Um Maschinen (wie Fahr- und Flugzeuge) effizienter nutzen zu können, wird deren Fähigkeit, autonom zu agieren und zu entscheiden, immer wichtiger [1]. Durch die zunehmende Verbreitung solcher autonom agierender Maschinen (auch Agenten genannt) entstehen immer mehr Situationen, in denen diese in naher Umgebung von Menschen agieren. Dabei existieren verschiedene Arten von Agenten, so zum Beispiel Belief-Desire-Intention-Agenten, die starke Annahmen über die Umwelt enthalten [2], oder Reinforcement-Learning-Agenten, die als flexibler angesehen werden [3]. Um eine sichere Zusammenarbeit und Interaktion mit Menschen zu gewährleisten, muss für jede Art von Agent ethisches Verhalten implementiert und bestenfalls garantiert werden. Am ethischen Verhalten der Agenten besteht von vielen Seiten Interesse:

Zuvorderst Menschen, die mit oder in der Umgebung von autonomen Agenten arbeiten. Diese müssen über die Entscheidungen, die ein Agent in einer bestimmten Situation trifft, informiert sein, da das ethische Verhalten eines Agenten direkt mit der Sicherheit der ihn umgebenden Menschen verbunden ist.

Dies führt neben weiteren Faktoren dazu, dass auch für die Hersteller autonomer Agenten (wie zum Beispiel Produzenten von Fahr- und Flugzeugen) deren ethisches Verhalten eine wichtige Rolle spielt. Hier ist insbesondere die Rolle der Garantie der ethischen Korrektheit des Agenten hervorzuheben: Diese Garantie wird von potentiellen Kunden eingefordert und kann einen wesentlich Kaufgrund darstellen, auch, weil ethische Dilemmata wie das Trolley-Problem sehr bekannt sind. Zudem vereinfacht sie den Zulassungsprozess enorm, da Fehler nicht mehr im Design eines Agenten auftreten können. Dies führt ebenfalls zu einer rechtlichen Absicherung eines Teiles des Produktionsprozesses. Ein weiterer wichtiger Punkt sind die Kosten des Design- und Implementierungsprozesses. Um diese zu verringern, sollte ein möglichst großer Teil hiervon gut überprüfbar und automatisch stattfinden.

Für die zuvor genannten Gruppen bietet die formale Verifikation ethischer Entscheidungen eine Lösung, da diese automatisch ein gegebenes Modell überprüfen und Garantien für formale (ethische) Eigenschaften liefern kann [2]. Eine Methode der formalen Verifikation, das Model Checking, wird in diesem Bericht beschrieben.

Hierfür werden in Sektion II die benötigten Grundlagen erklärt, in III die Besonderheiten bei der Verifikation ethischer Regeln beschrieben und in IV ein möglicher Implementierungsansatz für Belief-Desire-Intention-Agenten vorgestellt, der in V an Beispielen veranschaulicht wird. In VI wird dieser Ansatz diskutiert und Erweiterungsmöglichkeiten in Richtung Reinforcement Learning aufgezeigt. Hierbei wird grob dem Aufbau einer Value Proposition gefolgt: Zuvor wurden bereits die Stakeholder beschrieben, im Folgenden wird ein Lösungsansatz und dessen Vor- und Nachteile präsentiert.

II. Grundlagen

A. Software-Agenten

Um die Komplexität und Wartbarkeit komplexer Software-Systeme zu verbessern, ist es möglich, diese Software agentenbasiertes umzusetzen. Hierbei werden *Agenten* als einzelne Softwareeinheiten mit definiertem Ziel und autonomen Verhalten eingesetzt. Diese können miteinander agieren, um so ein Gesamtziel zu erreichen. Die auf diese Weise entstehenden Agenten können damit auf bisher nicht bekannte Situationen reagieren und führen zu einem modularen Aufbau des gesamten Software-Systems [4], [5]. Es existieren verschiedene Arten, diese Agenten umzusetzen, wie zum Beispiel Belief-Desire-Intention-Agenten oder Agenten basierend auf Reinforcement Learning.

1) Belief-Desire-Intention-Agent: Belief-Desire-Intention-Agenten (BDI-Agenten) sind menschlichen Entscheidungsprozessen nachempfunden. Namensgebend sind ihre drei Grundbestandteile: *Beliefs*, die Annahmen, die einem Agenten über seine Umwelt zur Verfügung stehen. Diese Annahmen beziehen sich auch auf die erwarteten Ergebnisse der Aktionen des Agenten. Da das Umfeld eines Agenten typischerweise nicht mit einer einzigen Sensorwahrnehmung erfasst werden kann, werden die *Beliefs* mit neuen Sensorinformationen angepasst, um so ein möglichst aktuelles Modell der Umgebung zu erhalten. *Desires* stellen die Ziele und damit verbundene Prioritäten oder Ergebnisse dar. Diese Ziele müssen untereinander nicht konsistent oder realistisch sein. *Intentions* geben die im Moment aktiv verfolgten und somit notwendigerweise gleichzeitig erreichbaren Ziele an. Eine andere Beschreibung der mentalen Attribute *Beliefs, Desires, Intentions* sind informative, motivationale und abwägende Zustände [6].

Basierend auf diesen drei Komponenten findet ein Entscheidungsprozess statt, der den Agenten seinen Zielen näherbringen soll. Hierfür stehen vorgefertigte Aktionen zur Verfügung, die der Agent basierend auf seinem Zustand und einem ebenfalls vorgefertigten Entscheidungsprozess auswählt [7].

2) Reinforcement-Learning-Agent: Agenten basierend auf Reinforcement Learning (RL-Agenten) sind konzeptionell ähnlich aufgebaut wie BDI-Agenten: Ihre wesentlichen beschreibenden Elemente lassen sich in Eingänge (wie Sensordaten), Aktionen und Ziele unterteilen. Der wichtigste Unterschied zu BDI-Agenten besteht darin, wie diese Ziele erreicht werden können. Während BDI-Agenten nach einem vorgefertigten Entscheidungsprozess aus einem Satz an vorgefertigten Aktionen auswählen, über deren Ergebnis sie Annahmen erhalten haben, lernen *RL-Agenten* basierend auf Interaktionen mit ihrer Umgebung [8]. Hierbei wird bei den gängigen Ansätzen versucht, einen Gewinn der Form

$$R = \sum_{t=0}^{inf} \gamma^t * r_t \tag{1}$$

zu maximieren. Dabei ist R der kumulative Gewinn aller kommender Zeitschritte t und $\gamma \in [0, 1)$ der Diskontierungsfaktor der in Zukunft erwarteten Gewinne r_t. Indem der Agent mit der Umwelt interagiert, lernt er, in welcher Situation welche Aktionen den meisten Gewinn bringen. Dabei muss - je nach verwendeter Methode - eine Aktion nicht direkt Gewinn bringen, dieser kann auch erst nach weiteren Aktionen oder Zeitschritten eintreten [9].

B. Formale Verifikation

Als *formale Verifikation* versteht man eine Gruppe an Verfahren, die verwendet werden können, um bestimmte Eigenschaften eines Systems nachzuweisen [10]. Dafür müssen die gewünschten Eigenschaften des Systems in formal überprüfbarer Spezifikation vorliegen. In der formalen Verifikation sollen die Vorteile von Tests per Simulation und manuellen mathematischen Beweisen verbunden und deren Nachteile ausgeglichen werden:

Manuelle Beweise sind sowohl sehr aufwändig, als auch für große Systeme sehr fehleranfällig, da zu leicht Fehler entstehen und übersehen werden. Dafür garantieren fehlerfreie manuelle Beweise die Korrektheit des Ergebnisses.

Tests per Simulation sind verhältnismäßig einfach durchzuführen, geben allerdings wenig Aufschluss über die Vollständigkeit des Ergebnisses.

Im Gegensatz dazu wird bei der formalen Verifikation per Computer ein System mit mathematischen Methoden auf die gegebenen Spezifikationen überprüft. Damit lassen sich automatisch und vollständig Eigenschaften eines Systems überprüfen. Für komplexe Systeme steigt hier der Aufwand allerdings stark an, wofür mehrere Anpassungen formaler Verifikationsmethoden existieren [11]. Die zwei meistverwendeten formalen Verifikationsmethoden sind Model Checking und Theorem Proving, wobei wir uns auf Model Checking beschränken.

Beim *Model Checking* werden alle möglichen Ausführungen eines Systems auf eine formale Eigenschaft überprüft. Hierfür wird - basierend auf einem Agenten und dessen Umgebung - automatisiert ein Zustandsgraph aufgebaut. Zur Untersuchung dieses Graphen sind gut fundierte Methoden bekannt. Die gewünschte Eigenschaft wird mit Operatoren ausgedrückt, die angeben, ob diese Eigenschaft irgendwann in der Zukunft, zu einem bestimmten Zeitpunkt oder immer erfüllt sein muss [2], [10]. *Model Checking* ist vollständig und kann somit die Korrektheit des Ergebnisses garantieren. Sollte eine Eigenschaft nicht erfüllt sein, wird bei gängigen Implementationen ein Gegenbeispiel erzeugt [12].

III. Besonderheiten bei der Verifikation ethischer Regeln

Die besondere Schwierigkeit bei der formalen Verifikation ethischer Probleme ist die notwendige exakte Spezifikation der zu prüfenden Eigenschaft, da ethische Prinzipien selten in exakter und an die Situation angepasster ausführbarer Form vorliegen. Dies wird in Sektion VI in der Diskussion des vorgestellten Implementierungsansatzes weiter ausgeführt. Wir nehmen im Folgenden an, dass die ethischen Regeln exakt vorgegeben sind (zum Beispiel von Seite des Gesetzgebers). Doch selbst mit einem ethischen Regelwerk können noch Situationen auftreten, in denen nicht alle ethischen Regeln gleichzeitig befolgbar sind. Auch für diese Situation sollte vorgegeben sein, wie zu handeln, bzw. welche Regel in welcher Situation zu brechen ist. Diese Annahmen sind in bestimmten Kontexten schon erfüllt, so ist in gewissen medizinischen Situation (wie zum Beispiel bei Formen der Sterbehilfe) eine klare Handlungsanweisung gegeben, in welcher Situation welche ethischen Regeln gebrochen werden dürfen. Falls Regeln verletzt werden müssen, nehmen wir im Folgenden an, dass gegeben ist, welche Regeln als wichtiger zu beachten gelten als andere und dass möglichst wenige und unwichtige Regeln verletzt werden sollen.

IV. Beispielhafter Implementierungsansatz eines ethischen BDI-Agenten

Mit den bisher vorgestellten Grundlagen lässt sich ein beispielhafter Implementierungsansatz eines BDI-Agenten erstellen. Hierfür werden die für den Ansatz gültigen Annahmen beschrieben und darauf basierend eine Ordnung ethischer Regeln entwickelt.

A. Annahmen

Im Folgenden werden diese (zum Teil bereits bekannten) Annahmen getroffen:

- Es ist ein eindeutiges ethisches Regelwerk gegeben. Die darin enthaltenen Regeln sind konkret, anwendbar und geordnet.
- Die Komplexität der untersuchten Systeme ist begrenzt, da Model Checking ein vollständiges Verfahren ist und große Systeme zu einer Zustandsexplosion und somit einer deutlich längeren Laufzeit führen.
- Dem Agenten stehen vorgefertigte Subroutinen als Aktionen zur Verfügung. Für diese Routinen ist bekannt, wie sie mit der Umwelt, den Zielen des Agenten und den gegebenen Regeln interagieren.
- Es wird unterschieden, wie oft eine Regel gebrochen wird, allerdings nicht, wie deutlich dies geschieht. Beispielsweise spielt es für die Regel "Keine Sachbeschädigung" keine Rolle, ob ein anderer physischer Agent leicht beschädigt oder vollständig zerstört wird. Werden allerdings zwei Agenten beschädigt, wird dies als zweifache Übertretung der Regel "Keine Sachbeschädigung" gewertet [2].

B. Entwicklung einer Ordnung ethischer Regeln

Um Model Checking anwenden zu können, müssen die gewünschten Eigenschaften des Agenten formal spezifiziert werden. In [2] werden dafür für folgende Konzepte Definitionen vorgeschlagen:

- Definition 1: Es wird eine ethische Regel E_ϕ definiert.
- Definition 2: Basierend darauf wird eine ethische Policy definiert. Diese Policy enthält eine Menge an ethischen Regeln und eine totale Ordnung auf diesen, die angibt, wie wichtig welche ethische Regel ist. Zusätzlich ist immer die leere ethische Regel $E_{\phi_\emptyset}$ enthalten. Diese ist erfüllt, wenn keine ethische Regel verletzt wird. Wenn nun in der Ordnung definiert wird, dass die leere ethische Regel die wichtigste Regel ist, wird vom Agenten immer bevorzugt, keine ethische Regel zu brechen.
- Definition 3: Weiterhin wird eine Notation definiert, die angibt, dass das Ausführen einer Aktion a die ethische Regel E_ϕ bricht.
- Definition 4: Als letztes wird definiert, wie die Pläne des Agenten basierend auf der ethischen Policy geordnet werden. Dabei wird angenommen, dass ein Plan p_i besser als ein Plan p_j ist, wenn p_i weniger wichtige ethische Regeln als p_j bricht oder gleich wichtige, dafür allerdings weniger davon.

Model Checking benötigt eine endliche Menge, über die iteriert werden kann. Deshalb müssen den Plänen des Agenten, die aufgrund ihrer beliebigen Anzahl nicht iterierbar sind, ethische Policys zugeordnet werden, von denen nur eine endliche Menge existiert.

V. Beispiele

Basierend auf dem vorgestellten Implementierungsansatz kann nun eine Sprache entwickelt werden, mit der sich folgende Beispiele einer unbemannten Drohne auswerten lassen [2]. Mit dem ersten Beispiel soll die grundsätzliche Funktionsweise einer Entscheidung verdeutlicht werden. Im zweiten Beispiel wird gezeigt, was passiert, wenn der Agent Aktionen versucht, die nicht zum gewünschten Ergebnis führen.

A. Auffahren auf die Rollbahn

Im ersten Beispiel fährt die Drohne auf ein Rollfeld, um von dort einen Flug zu starten. In diesem Kontext gelten für die unbemannte Drohne diese *ethischen Regeln*:

1) Kollidiere nicht mit bemannten Flugzeugen.
2) Verletze keine Menschen.
3) Beschädige keine anderen Dinge.
4) Beschädige dich nicht selbst.

Diese sind ihrer Wichtigkeit nach geordnet, jede Regel ist wichtiger als die Regel unter ihr. Um den Entscheidungsprozess des Agenten zu evaluieren, können verschiedene Eingänge und Umgebungen vorgegeben werden. Während dem Auffahren auf die Rollbahn detektiert die Drohne nun ein bemanntes Flugzeug, das sich in der geplanten Bahn befindet und zu nahe ist, um noch rechtzeitig bremsen zu können. Die *Umgebung der Drohne* kann auf wesentliche Elemente reduziert so beschrieben werden:

- Links der Landebahn befinden sich Flughafenlichter.
- Geradeaus befindet sich das bemannte Flugzeug.
- Rechts der Rollbahn befinden sich Flughafenlichter und ein Mitarbeiter des Flughafens.

Als Reaktion auf die unerwartete Situation ergeben sich folgende *Aktionen und deren Auswirkung*:

- Ausweichen nach links: Sach- und Selbstbeschädigung
- Ausweichen nach rechts: Verletzung einer Person und Sach- und Selbstbeschädigung
- Kurs beibehalten: Kollision mit einem bemannten Flugzeug und Sach- und Selbstbeschädigung

Aus der Anschauung wird deutlich, dass immer die Option "Ausweichen nach links" gewählt werden sollte, da diese die schwächsten ethischen Regeln bricht. Dies lässt sich für dieses Beispiel über Model Checking nachweisen und somit garantieren.

B. Ausweichen im Luftraum

In diesem Beispiel befindet sich die Unbemannte Drohne im Flug. In der Luft wird nun ein entgegenkommendes Flugzeug detektiert. Da sich somit der Kontext der ethischen Entscheidung ändert, ändern sich auch die in dieser Situation geltenden *ethischen Regeln*:

1) Kollidiere nicht mit anderen Flugzeugen.
2) Kollidiere nicht mit Objekten auf dem Boden.
3) Vollende den Flug zum Zielort.
4) Bleibe über der definierten Distanz zum Boden.
5) Drehe nach rechts ab.

Diese sind wieder nach ihrer Wichtigkeit sortiert. Die letzte Regel ergibt sich aus der Konvention, dass Piloten in dieser Situation nach rechts abdrehen sollten, um so Kollisionen zu vermeiden. Als *Aktionen* der Drohne ergeben sich dann:

- Kurs beibehalten: Kollision mit bemanntem Flugzeug
- Flughöhe senken: Kollision mit Objekten auf dem Boden, Distanz zum Boden zu gering
- Ausweichen nach links: Konvention (Regel 5) gebrochen
- Ausweichen nach rechts: keine Regel gebrochen
- Breche Flug ab und kehre zum Startflughafen zurück: Flug wird nicht vollendet

Die Informationen der Drohne sind in diesem Szenario so gewählt, dass Ausweichmanöver nicht erfolgreich sind. Somit versucht die Drohne erst, nach rechts abzudrehen, da dies alle ethischen Regeln erfüllt. Diese Aktion ist nicht erfolgreich und das entgegenkommende Flugzeug befindet sich noch auf Kollisionskurs, deshalb wird sie aussortiert. Als beste Aktion bleibt dann das Ausweichen nach links, das ebenfalls fehlschlägt. An diesem Punkt entscheidet die Drohne, den Flug abzubrechen und zum Startflughafen zurückzukehren, da dies die bestmögliche ethische Entscheidung ist, die noch zur Verfügung steht. So lässt sich in jedem Schritt des Entscheidungsprozesses über Model Checking nachweisen, dass die Drohne die bestmögliche ethische Entscheidung trifft.

C. Erweiterungen der Beispiele

Diese Beispiele lassen sich über randomisierte Elemente erweitern. So kann zum Beispiel angenommen werden, dass Pläne zur Verfügung stehen, die jede beliebige Kombination an ethischen Regeln brechen. Von dieser Menge kann eine zufällige Untermenge ausgewählt werden, auf der Model Checking angewendet werden kann, um nachzuweisen, dass immer die Pläne ausgewählt werden, die den ethischen Regeln entsprechen.

Eine andere Erweiterung stellt die Randomisierung des Eingangs des Agenten dar. So kann in Beispiel V-B das Verhalten des entgegenkommenden Flugzeuges in jedem Zeitpunkt zufällig gewählt werden, um so zu überprüfen, ob der Agent in jedem Zeitschritt den besten ethischen Plan auswählt.

VI. Diskussion des gewählten Ansatz und mögliche Verbesserungen

Im Folgenden wird der zuvor vorgestellte Implementierungsansatz diskutiert. Zu Beginn wird der - für die in der Einführung beschriebenen Stakeholder - entstehende Wert verdeutlich. Dann werden generelle Probleme bei ethischen Entscheidungen und Schwierigkeiten bei der Wahl einer Ordnung ethischer Regeln erläutert. Anschließend wird beschrieben, welche Möglichkeiten existieren, die formale Verifikation für flexiblere Agenten basierend auf Reinforcement Learning zu erweitern.

A. Vorteile des gewählten Ansatzes

Es wurde ein Implementierungsansatz für das ethische Verhalten eines Belief-Desire-Intention-Agenten vorgestellt. Dabei wurde ein exaktes ethisches Regelwerk in Kombination mit Model Checking verwendet, um dem Agenten gewünschte ethische Regeln vorzugeben und eine Garantie für die bestmögliche Einhaltung dieser Regeln zu erhalten. Diese Garantie wird automatisiert überprüft, somit kann die benötigte Mehrleistung gering gehalten werden. Für die Hersteller autonomer Agenten bietet die garantierte Korrektheit des ethischen Verhaltens mehrere wesentliche Vorteile:

In ethischen Dilemmata (wie in Sektion VI-B näher beschrieben) lässt sich bei fehlerfreier Funktion des Agenten ethisches Verhalten garantieren. Diese ethischen Dilemmata sind in verschiedener Form seit mehreren Jahrzehnten der Öffentlichkeit bekannt und werden so bei der Markteinführung (zum Beispiel von autonomen Fahrzeugen) eine wichtige Rolle spielen. Autonome Agenten können sich in ethischen Dilemmata befinden und müssen sowohl für die Zulassung, als auch als Kaufgrund für Endverbraucher, nachweisen können, wie sie in welcher Situation reagieren. Mit der formalen Verifikation kann hier eine Garantie des gewünschten Verhaltens gegeben werden, um somit rechtliche Probleme bei der Zulassung und Misstrauen bei möglichen Kunden zu umgehen.

Ein weitere Vorteil entsteht durch die automatisierte Überprüfung des Agenten-Designs. Mit sehr geringer Ingenieursleistung kann eine Haftung für unethisches Verhalten eines Agenten ausgeschlossen werden. Diese Automatisierung führt außerdem dazu, dass Fehler im Verifikationsprozess eines Agenten verhindert werden. Ethisch falsche Entscheidungen können nur aus Gründen, die außerhalb des Einflusses des Software-Herstellers liegen, geschehen, so zum Beispiel Hardwarefehler oder bewusster Missbrauch eines Agenten.

Die wesentlichen Vorteile des vorgestellten Ansatzes entstehen somit aus der Garantie, dass ein Agent stets den exakt vorgegebenen ethischen Regeln folgt. Diese Regeln zu definieren birgt allerdings Probleme, wie im Folgenden beschrieben wird.

B. Probleme bei ethischen Entscheidungen

Das ethische Verhalten eines Agenten lässt sich (wie in Sektion V) durch dessen Reaktion in ethischen Dilemmata überprüfen [13]. Eines der bekanntesten Dilemmata stellt das sogenannte Trolley-Problem dar, das in vielen verschiedenen Ausführungen Verwendung findet und im Original auf [14] basiert. Dabei wird folgende Situation beschrieben: Die handelnde Person befindet sich an der Weiche eines Bahngleises und sieht einen Zug näherkommen. An der Weiche teilt sich die Bahnlinie in zwei weitere Gleise auf, auf einem befinden sich fünf Personen, auf dem anderen nur eine. Diese Personen können die Strecke nicht rechtzeitig verlassen, um dem Zug auszuweichen. Ohne Handlung werden die fünf Personen überfahren. Durch das Umlegen eines Schalters wird der Zug auf das andere Gleis mit der einzelnen Person umgeleitet.

Als Beispiel für Vor- und Nachteile der bisher genutzten ethischen Regeln bieten sich zwei Abwandlungen des Trolley-Problems an:

Ein autonomes Fahrzeug stellt zu spät fest, dass sich Menschen auf und neben der eigenen Fahrbahn befinden. Auf der Fahrbahn sind fünf Personen, daneben eine. Der Agent muss sich nun entscheiden, ob er den Personen auf der Fahrbahn ausweicht, damit aber die Person neben der Fahrbahn gefährdet, oder ob er seinen Kurs beibehält und die Personen auf der Fahrbahn gefährdet. Unter der Annahme, dass der Agent für diesen Fall nur die relevante Regel "Verletze keine Menschen" enthält und das zuvor vorgestellte Konzept verwendet wird, kommt der Agent zu einer eindeutigen Entscheidung. Bei beiden Aktionen (Ausweichen oder Kurs beibehalten) wird die Regel "Verletze keine Menschen" verletzt, allerdings beim Ausweichen einfach und beim Beibehalten des Kurses fünffach. Somit würde sich der Agent hier immer dafür entscheiden, auszuweichen. Für diese Situation scheint das zuvor vorgestellte Konzept eine - großteils als besser angesehene [15] - Lösung zu garantieren.

Ein Fehler des Konzeptes zeigt sich allerdings in folgender Situation: Ein autonomes Fahrzeug stellt zu spät fest, dass sich eine Person auf der Strecke befindet, Ausweichmöglichkeiten stehen nicht zur Verfügung. Als Aktionen kann der Agent bremsen oder die Geschwindigkeit beibehalten. Vorausgesetzt der Agent folgt der Regel "Verletze keine Menschen" und es wird nicht weiter überprüft, in welcher Schwere eine Aktion diese Regel bricht, sind beide zuvorgenannten Aktionen (Bremsen oder Geschwindigkeit beibehalten) für den Agenten gleichbedeutend. Ein Mensch in der gleichen Situation würde sich allerdings für das Bremsen entscheiden, um so den entstehenden Schaden zu verringern. Dieser Unterschied lässt sich entweder auf einen Fehler des verwendeten Implementierungsansatzes rückführen oder auf eine ungewünschte Folge ungenau definierter ethischer Regeln.

C. Wahl einer Ordnung ethischer Regeln

Bei dem zuvor vorgestellten Beispiel zeigt sich, dass für ethische Entscheidungen betrachtet werden muss, in welcher Deutlichkeit welche Regel gebrochen wird und wie wichtig die exakte Formulierung der gegebenen Regeln für das Verhalten des Agenten ist. Zusätzlich kann es durch den Nichtdeterminismus der realen Welt erforderlich sein, bei der Entscheidung einzubeziehen, wie wahrscheinlich eine Regel durch eine gewisse Aktion gebrochen wird. Außerdem gelten diese erstellten Regeln typischerweise in genau einem Kontext. Dies führt dazu, dass für jeden möglichen Kontext des Agenten ein Regelwerk ausgearbeitet werden muss. Nur so kann ein ethischer Agent erhalten werden, der in jedem Kontext richtig handelt. Hier stellt sich allerdings die Frage, wie exakt die zu untersuchenden Kontexte differenziert werden: Ist eine Situation wie zuvor mit fünf Personen auf und einer Person neben der Strecke ausreichend oder sollte dabei zum Beispiel noch betrachtet werden, ob diese Personen sich dort aufhalten sollten, wo sie im Moment sind? So könnte zum Beispiel die Entscheidung entstehen, dass der Kurs beizubehalten ist, da sich die Person neben der Strecke befinden darf, die Personen auf der Fahrbahn allerdings nicht. Die Gewichtung der Regeln in einem Kontext ist zudem subjektiv und je nach Kultur und persönlicher Präferenz unterschiedlich.

Diese Punkte führen dazu, dass es sehr schwierig ist, für einen allgemeinen ethischen Agenten ein Regelwerk zu erstellen, das in jeder Situation ethisch korrektes Verhalten erzeugt. Dieses Regelwerk müsste in seiner Vollständigkeit mit formaler Verifikation überprüft werden, um eine Garantie für die ethische Korrektheit des Agenten zu erhalten.

D. Reinforcement Learning

Bisher wurde davon ausgegangen, dass die Ergebnisse einzelner Aktionen dem Agenten stets bekannt sind. Damit kann für eine Aktion bestimmt werden, welche ethischen Regeln durch sie verletzt werden und es kann eine Ordnung der zu wählenden Aktionen erstellt werden. Bei einem Agenten basierend auf Reinforcement Learning wird durch die Interaktion mit der Umgebung gelernt. Basierend auf dem für eine Aktion erhaltenen Gewinn wird dann bewertet, ob eine Aktion in einer bestimmten Situation als nützlich angesehen wird. Dabei kann eine Aktion auch erst nach mehreren Zeitschritten zu einem Gewinn für den Agenten führen. Dies führt dazu, dass der Agent nicht mit Wissen über die Umgebung ausgestattet sein muss, wodurch die Verifikation bestimmter Eigenschaften wie zum Beispiel ethischem Verhalten erschwert wird. Zusätzlich muss durch den verzögerten Gewinn einer Aktion nicht nur ein Zustand oder eine Aktion, sondern eine Sequenz an Zuständen und Aktionen betrachtet werden [12]. Um dennoch bei RL-Agenten ethisches Verhalten zu erzeugen oder zu verifizieren, existieren verschiedene Ansätze, wie zum Beispiel den Agenten ethische Regeln lernen zu lassen oder Anpassungen formaler Verifikationsmethoden.

1) Lernen ethischer Regeln: Ein möglicher Ansatz, bei einem RL-Agenten für ethisches Verhalten zu sorgen, ist es, den Agenten dieses lernen zu lassen. Hierfür werden nicht ethische Aktionen durch hohe Kosten bzw. einen negativen Gewinn "bestraft", um dem Agenten beizubringen, diese Aktionen zu vermeiden. Dieser Ansatz ist gut vorstellbar und bietet den Vorteil, dass kein festes Regelwerk für den Agenten gegeben sein muss. Allerdings ist das gelernte Verhalten des Agenten von Faktoren wie der Implementierung des Agenten und der Kostenfunktion abhängig, die auch wieder auf ethischen Grundsätzen basierend erstellt werden muss. Diese Implementierung ist gegenüber kleinen Fehlern sehr sensitiv, so kann ein Fehler in der Kostenfunktion dazu führen, dass die ethischen Regeln nicht oder unzureichend gelernt werden. Außerdem muss sichergestellt sein, dass der Agent im Training allen Situationen mit ethischen Entscheidungen hinreichend oft ausgesetzt ist, ansonsten ist nicht absehbar, wie der Agent in den realen Situationen reagiert. Diese Agenten sind schwierig auf ihr ethisches Verhalten zu überprüfen und werden oft mit ethischen Dilemmata (siehe auch Sektion VI-B) getestet, um so das Verhalten des Agenten bewerten zu können.

2) Formale Verifikation für RL-Agenten: Für einen einzelnen Zustand eines Agenten ist es recht einfach, nachzuweisen, dass der Agent keine unethische Aktion wählt [16]. Einem RL-Agenten stehen allerdings typischerweise so viele Zustände und Aktionen zur Verfügung, dass diese - wie für Model Checking notwendig - nicht mehr alle untersucht werden können. Es existieren verschiedene Ansätze, um die formale Verfikation an RL-Agenten anzupassen.

Ein Ansatz definiert eine Sicherheitsfunktion, mit der über semi-formale Verifikation die Sicherheit des (ethischen) Verhaltens des Agenten bewertet werden kann [17]. Hierbei können allerdings keine Garantien für die Korrektheit des ethischen Verhaltens gegeben werden.

Eine weitere Methode der semi-formalen Verifikation - begrenztes Model Checking - kann verwendet werden, um automatisiert einen RL-Agenten auf bestimmte Eigenschaften zu überprüfen [12]. Dafür wird ein geringerer Rechenaufwand benötigt als bei vollständigem Model Checking. Falls eine gewünschte Eigenschaft nicht zutrifft, können Gegenbeispiele in Form von Zustandssequenzen erzeugt werden. Dies führt dazu, dass automatisiert gezeigt werden kann, welche Eigenschaften nicht zutreffen. Allerdings existiert auch bei diesem Verfahren keine Garantie, dass eine Eigenschaft immer zutrifft, falls kein Gegenbeispiel gefunden wird.

Ein weiterer Ansatz kombiniert Theorem Proving - eine andere Methode der formalen Verifikation - mit der Überprüfung des Agenten zur Laufzeit, um so Garantien für das Verhalten des Agenten zu erhalten [18]. Diese Garantien gelten allerdings nur für akkurate Modelle der Umgebung und des Agenten, wobei in Experimenten auch für Abweichungen der Umgebung vom Modell sichere Ergebnisse erzielt wurden.

VII. Fazit

Durch die zunehmende Verwendung und Interaktion mit Menschen wird das ethische Verhalten von Software-Agenten und dessen Verifikation immer wichtiger. Hierfür bietet die formale Verifikation allgemein und Model Checking im Speziellen einen möglichen Lösungsansatz. Dieser zeichnet sich durch die manuelle Spezifikation exakter Regeln aus, die anschließend automatisch überprüft werden können. Für Belief-Desire-Intention-Agenten lässt sich so Verhalten erzeugen, das - durch die Vollständigkeit von Model Checking garantiert - gegebenen ethischen Regeln folgt. Dabei kann durch die automatische Überprüfung der Aufwand des System-Designers gering gehalten werden, während das Design des Software-Agenten abgesichert wird. Das wesentliche Problem stellt hierbei die Spezifikation der ethischen Regeln und deren Ordnung dar, da diese subjektiv und für jede möglicherweise auftretende Situation zu definieren sind.

Die Methoden der formalen Verifikation lassen sich anpassen, um sie auf flexiblere Agenten wie Reinforcement-Learning-Agenten anwenden zu können. Insbesondere diese Entwicklungsmöglichkeit erscheint zukunftsweisend, da RL-Agenten flexibel ihre Umgebung erlernen können und für ethisches Verhalten nicht zwangsweise ein exaktes Regelwerk benötigt wird. Wenn dabei garantiert werden kann, dass ein Agent nur sicheres Verhalten lernen und ausführen kann, lässt sich eine Vielzahl verschiedener Probleme lösen, während gleichzeitig die Sicherheit aller Personen in der Umgebung des Agenten gewährleistet wird.

References

[1] Wilko Schwarting, Javier Alonso-Mora, and Daniela Rus. Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, 1(1):187–210, 2018.

[2] Louise Dennis, Michael Fisher, Marija Slavkovik, and Matt Webster. Formal verification of ethical choices in autonomous systems. *Robotics and Autonomous Systems*, 77:1 – 14, 2016.

[3] Ah-Hwee Tan, Yew-Soon Ong, and Akejariyawong Tapanuj. A hybrid agent architecture integrating desire, intention and reinforcement learning. *Expert Systems with Applications*, 38(7):8477–8487, 2011.

[4] Nicholas R. Jennings. On agent-based software engineering. *Artificial Intelligence*, 117(2):277 – 296, 2000.

[5] Michael Wooldridge. Agent-based software engineering. *IEE Proceedings-software*, 144(1):26–37, 1997.

[6] Michael Georgeff, Barney Pell, Martha Pollack, Milind Tambe, and Michael Wooldridge. The belief-desire-intention model of agency. In *International workshop on agent theories, architectures, and languages*, pages 1–10. Springer, 1998.

[7] Anand Srinivasa Rao and M. Georgeff. Bdi agents: From theory to practice. In *ICMAS*, 1995.

[8] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

[9] Marco Wiering and Martijn Van Otterlo. *Reinforcement learning*, volume 12. Springer, 2012.

[10] Edmund M. Clarke. *The Birth of Model Checking*, pages 1–26. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[11] Willem Visser, Klaus Havelund, Guillaume Brat, SeungJoon Park, and Flavio Lerda. Model checking programs. *Automated software engineering*, 10(2):203–232, 2003.

[12] Yafim Kazak, Clark Barrett, Guy Katz, and Michael Schapira. Verifying deep-rl-driven systems. In *Proceedings of the 2019 Workshop on Network Meets AI ML*, NetAI'19, page 83–89, New York, NY, USA, 2019. Association for Computing Machinery.

[13] Vivek Nallur. Landscape of machine implemented ethics. *Science and Engineering Ethics*, 26(5):2381–2399, Jul 2020.

[14] Philippa Foot. The problem of abortion and the doctrine of double effect. 1967.

[15] Jean-François Bonnefon, Azim Shariff, and Iyad Rahwan. The social dilemma of autonomous vehicles. *Science*, 352(6293):1573–1576, 2016.

[16] Perry Van Wesel and Alwyn E Goodloe. Challenges in the verification of reinforcement learning algorithms. 2017.

[17] Davide Corsi, Enrico Marchesini, and Alessandro Farinelli. Evaluating the safety of deep reinforcement learning models using semi-formal verification. *arXiv preprint arXiv:2010.09387*, 2020.

[18] Nathan Fulton and André Platzer. Safe reinforcement learning via formal methods: Toward safe control through proof and learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.