HAL
archives-ouvertes.fr

Cadre Déclaratif Modulaire pour Représenter et Appliquer des Principes Éthiques

Fiona Berreby, Gauvain Bourgne, Jean-Gabriel Ganascia

▶ **To cite this version:**

Fiona Berreby, Gauvain Bourgne, Jean-Gabriel Ganascia. Cadre Déclaratif Modulaire pour Représenter et Appliquer des Principes Éthiques. Journées d'Intelligence Artificielle Fondamentale, Jul 2017, Caen, France. Journées d'Intelligence Artificielle Fondamentale, 2017. <hal-01564673>

HAL Id: hal-01564673
https://hal.sorbonne-universite.fr/hal-01564673

Submitted on 19 Jul 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Cadre Déclaratif Modulaire pour Représenter et Appliquer des Principes Éthiques

Fiona Berreby[1] **Gauvain Bourgne**[1] **Jean-Gabriel Ganascia**[1]

[1] CNRS & Sorbonne Universités, UPMC Université Paris 06, LIP6 UMR 7606,
4 place Jussieu 75005 Paris,France
fiona.berreby@lip6.fr gauvain.bourgne@lip6.fr jean-gabriel.ganascia@lip6.fr

Abstract

Cet article examine l'utilisation de langages de haut niveau dans la conception d'agents autonomes éthiques. Il propose un cadre logique nouveau et modulaire pour représenter et raisonner sur une variété de théories éthiques, sur la base d'une version modifiée de l'Event Calculus, implémentée en Answer Set Programming. Le processus de prise de décision éthique est conçu comme une procédure en plusieurs étapes, capturée par quatre types de modèles interdépendants qui permettent à l'agent d'évaluer son environnement, de raisonner sur sa responsabilité et de faire des choix éthiquement informés. Notre ambition est double. Tout d'abord, elle est de permettre la représentation systématique d'un nombre illimité de processus de raisonnements éthiques, à travers un cadre adaptable et extensible en vertu de sa hiérarchisation et de sa syntaxe standardisée. Deuxièmement, elle est d'éviter l'écueil de nombreux travaux d'éthique computationnelle qui directement intègrent l'information morale dans l'engin de raisonnement général sans l'expliciter -alimentant ainsi les agents avec des réponses atomiques qui ne représentent pas la dynamique sous-jacente. Nous visons à déplacer de manière globale le fardeau du raisonnement moral du programmeur vers le programme lui-même.

1 Introduction

L'étude de la morale d'un point de vue computationnel a attiré l'intérêt croissant de chercheurs en intelligence artificielle[2]. En effet, l'autonomie croissante des agents artificiels et l'augmentation du nombre de tâches qui leur sont déléguées nous incitent à aborder leur capacité à traiter les restrictions et les préférences éthiques, que ce soit dans leur propre structure interne ou pour des interactions avec des utilisateurs humains. Des domaines aussi variés que la santé ou le transport posent des problèmes éthiques qui sont en ce sens particulièrement pressants, car ils peuvent exiger des agents des prises de décisions dont les conséquences sont immédiates ou lourdes. L'éthique computationnelle peut aussi nous aider à mieux comprendre la morale et raisonner plus clairement sur les concepts éthiques qui sont employés dans des domaines philosophiques, juridiques et technologiques. Dans ce contexte, notre objectif est de fournir une architecture modulaire qui permette la représentation systématique et adaptable des principes éthiques. Pour ce faire, nous présentons un ensemble cohérent de modèles qui, ensemble, permettent à l'agent d'évaluer son environnement, d'intégrer des règles éthiques et déterminer à partir de la mise en œuvre de ces règles soit un plan d'action, soit une évaluation du comportement d'autres agents. Ceux-ci sont implémentés en Answer Set Programming [1], sur la base d'une version modifiée de l'Event Calculus. Ainsi, notre approche est une approche logique de l'éthique, qui existe en parallèle à d'autres telles que l'éthique par étude de cas ([3][2]), ou l'éthique par conception ([5]).

Nous avons choisi l'utilisation de la *logique non monotone* car son étude a été proposée comme moyen de gérer le genre de généralisations révocables qui caractérisent souvent le raisonnement de sens commun et qui sont mal capturées par les systèmes logiques classiques [17]. Le terme couvre une famille de cadres formels conçus pour appréhender le type d'inférence où aucune conclusion n'est définitive, mais reste ouverte à la modification à la lumière d'informations complémentaires. Ce type de raisonnement par défaut est constitutif du raisonnement éthique. Des facteurs tels que la présence d'options alternatives, de conséquences indirectes ou de circonstances atténuantes peuvent renverser le jugement éthique d'une action. En conséquence, les langages non monotones sont particulièrement adaptés à la modélisation du raisonnement éthique.

1. Pour une description de l'Answer Set Programming, voir [21].

L'article est structuré comme suit. Nous commençons par présenter les concepts philosophiques pertinents ainsi que les travaux connexes [Sect.2], puis nous présentons l'architecture du cadre [Sect.3]. Ensuite, nous définissons et discutons de chaque modèle [Sects.4-7], puis illustrons leur implémentation à l'aide d'un exemple jouet [Sect.8], et concluons [Sect.9].

2 Motivation

2.1 Théories Éthiques

L'étude de l'éthique est l'étude des croyances que les gens peuvent ou devraient avoir pour contrôler leur comportement. Une classification tripartite standard divise le champ entre la *méta-éthique*, qui concerne le statut ontologique des concepts éthiques, l'*éthique appliquée*, qui concerne l'application des règles morales à des environnements particuliers, et l'*éthique normative*, qui traite de la définition, de la comparaison et de l'explication de conceptions éthiques [10]. Le présent travail présente un intérêt pour l'éthique appliquée en ce sens qu'il présente un schéma de conception d'agents artificiels contraints par l'éthique qui peuvent agir dans une variété de domaines appliqués. Il présente aussi un intérêt pour l'éthique normative car son but est de modéliser les processus qui sous-tendent la prise de décision éthique normative, avec la possibilité de confronter des perspectives différentes. Il se concentre sur deux de ses principales branches : l'éthique conséquentialiste et l'éthique déontologique.

Le Bien et Le Juste

Les éthiques conséquentialistes s'articulent autour de l'idée que les actions doivent être évaluées en fonction de leur conséquences, et ne peuvent être justes ou injustes qu'en vertu de ce qu'elles produisent. Une action moralement juste est celle qui produit un bon, ou le meilleur, état de choses. Hors, afin de déterminer la justesse d'une action, les conséquentialistes doivent d'abord établir ce qui constitue un *bon* état de choses, c'est-à-dire déterminer ce qu'on appelle plus largement 'Le Bien' [1]. Cela leur permet ensuite d'affirmer que des actions font partie du 'Juste' dans la mesure où elles augmentent le Bien. Les théories conséquentialistes suivent donc, s'appuient et finalement dépassent, les théories du Bien. Les désaccords entre conséquentialistes sur ce qui constitue le Bien, ont engendré diverses traditions et doctrines conséquentialistes. L'*utilitarisme* voit le Bien résulter de la maximisation du bien-être collectif, l'*atruisme éthique* du bien-être des autres, l'*égoïsme éthique* de l'intérêt personnel, l'*utilitarisme des droits* du respect des droits individuels.

Les théories déontologiques (du grec *deon*, "devoir") prétendent que la valeur morale d'une action est déterminée (au moins en partie) par une caractéristique intrinsèque de l'action. Habituellement, cette caractéristique est une obligation ou une interdiction. Par exemple, une règle déontologique peut indiquer que le mensonge est contraire à l'éthique, ce qui implique que tout énoncé qui contient un mensonge est interdit. Parce qu'une action est jugée juste ou non en fonction de sa conformité avec une norme ou un devoir, son évaluation éthique est au moins en partie indépendante de ses conséquences. Le Juste est ici prioritaire sur le Bien : une action peut être injuste pour le déontologue même si elle maximise le Bien, et juste même si elle le minimise. Les tentatives de définition du Bien seront désormais appelées *théories du Bien*, et les tentatives de définition du Juste, qu'elles soient conséquentialistes ou déontologiques, seront appelées *théories du Juste*.

2.2 Travaux Connexes

Un certain nombre de travaux ont proposé des modèles informatiques de théories éthiques, dont la déontologie basée sur les devoirs et les règles [3] [5] [26], la déontologie de commande divine [10], le conséquentialisme [16] [20], ou l'instanciation de normes [29]. Il existe aussi des approches de vérification formelle [13][14]. Cependant, certains de ces modèles tendent à intégrer directement l'information éthiques au sein du processus de prise de décision de l'agent, sans pour autant générer un raisonnement moral. Bien qu'ils réussissent à exécuter des implémentations directes de restrictions uniques, ils ne fournissent pas une représentation explicite des relations de causalité ou des processus de pensée éthique, limitant ainsi leur applicabilité et leur portée. Par exemple, en utilisant une logique prospective, Pereira et al. [26] modélisent une règle déontologique qui prohibe le meurtre intentionnel par la règle '*falsum* ← *intentionalKilling*.' Or ils déterminent si '*intentionalKilling*' vaut pour une action en indiquant atomiquement si cette action l'implique, utilisant des règles de la forme '*intentionalKilling* ← *end(A,iKill(Y))*.' où A est l'action évaluée. Le problème avec cette approche est que l'évaluation éthique est *indiquée* par des énoncés spécifiques à l'action, plutôt qu'*extraite* par une forme de compréhension de l'environnement et des règles éthiques en place. Il n'y a pas de représentation de la causalité, de sorte que l'action et ses conséquences ne sont pas liées de dynamiquement ; leur relation est déclarée plutôt que déduite. Par conséquent, aucune notion de responsabilité éthique ne peut être élaborée sur cette base. En outre, les règles données manquant de puissance expressive, un nouveau programme est nécessaire pour modéliser chaque nouveau cas. Ces formalismes ne peuvent donc pas contraster différentes théories, ni expliciter leurs hypothèses.

Des travaux plus récents ont exploré de manière intégrale l'architecture des jugements éthiques [11] [9], représentant explicitement ces processus de raisonnement. Ce travail s'inscrit dans cette poursuite.

3 Schéma Structurel

3.1 Modèles et Modularité

La représentation explicite du raisonnement éthique permet à un agent d'informer son processus de prise de décision ou de juger du comportement des autres. Pour y parvenir, il 'teste' les actions possibles dans des simulations spécifiques afin d'évaluer leurs conséquences ou leur mérite éthique inhérent. Le résultat de la simulation donne alors un ensemble d'actions acceptables ou inacceptables, qui dicte le comportement à venir. Le cadre présenté ici est concerné par ce processus d'évaluation, plutôt que par ce que l'application de cette évaluation.

Le processus éthique est appréhendé comme une procédure en quatre étapes définie par quatre types de modèles interdépendants : un *modèle d'action*, un *modèle de causalité*, un *modèle du Bien*, et un *modèle du Juste*. Les deux premiers modèles produisent une compréhension entièrement non-éthique du monde, les deux suivants y superposent une compréhension éthique du monde. Le modèle d'action présenté ici, et qui constitue la base du cadre, est basé sur une version modifiée de l'Event Calculus à la manière de [9]. La situation est représentée en termes de *fluents*, des propriétés du monde variant dans le temps, et d'*évènements* qui modifient ces fluents. Nous définissons ces modèles ici, comme illustré dans la figure 1.

Un *modèle d'action* $\mathbb{A}$ permet à l'agent de représenter son environnement et les changements qui s'y déroulent. Il prend comme entrée un *ensemble d'actions effectuées*. Il est composé d'une *situation initiale* contenant les fluents vrais à T=0, une *spécification d'événements* contenant un ensemble d'événements et de relations de dépendance, et un *moteur d'événement* qui permet à la simulation d'évoluer. Il génère une *trace d'événements* de chaque simulation qui désigne pour chaque moment les événements qui s'y produisent et des fluents qui y sont vrais.

Un *modèle de causalité* $\mathbb{C}$ piste les conséquences des actions, rendant possible un raisonnement sur la responsabilité et l'imputabilité des agents. Il prend comme entrée la *trace d'événements* produite par le modèle d'action et une *spécification d'événements* contenant un ensemble d'événements et de relations de dépendance. Il est composé d'un *moteur causal* qui permet la création d'un arbre causal représentatif de la simulation. Il génère une *trace causale* de chaque simulation qui désigne pour chaque moment les liens de cause à effets qui existent entre événements et fluents.

Un *modèle du Bien* $\mathbb{G}$ donne une appréciation de la valeur éthique intrinsèque de finalités ou d'événements. Il est composé d'une *spécification de modalités*, d'une *spécification éthique d'événements* composée d'un ensemble d'événements et d'un ensemble de relations de dépendance éthique, et d'une ou plusieurs *théories du Bien*. Il génère une *évaluation du Bien*, évaluant les événements comme plus ou moins en accord avec le Bien.

Un *modèle du Juste* $\mathbb{R}$ détermine l'action la plus juste selon des circonstances données. Il prend comme entrée la *trace causale* produite par le modèle de causalité et, dans le cas où une théorie du Juste considérée contient des principes conséquentialistes, une *évaluation du Bien* produite par le modèle du Bien. Il est composé d'une ou plusieurs *théories du Juste* et, dans le cas où une théorie du Juste considérée contient des principes déontologiques, un *ensemble de spécifications déontologiques*. Il génère une *évaluation du Juste*, évaluant les actions comme plus ou moins en accord avec le Juste -et donc admissibles ou non.

Ces quatre types de modèles sont interdépendants à degrés variables. Les modèles du Bien et du Juste reposent toujours sur un modèle d'action et un modèle de causalité. Mais alors qu'un modèle de causalité est toujours nécessaire, la formulation particulière du moteur causal peut varier, par exemple pour représenter différentes définitions des causes et des conséquences. Parce que le moteur d'événement constitue la base du cadre, il est toutefois proposé qu'il soit unique et invariable. En ce qui concerne les modèles éthiques, avoir un modèle du Bien est nécessaire pour modéliser les théories du Juste qui sont conséquentialistes, ainsi que celles qui sont déontologiques lorsqu'elles comportent des contraintes conséquentialistes. Des interdépendances peuvent également exister au sein d'un type de modèle, en particulier dans le cas des théories du Juste qui peuvent faire appel l'une à l'autre. La hiérarchie bien définie entre les différents types de modèles donne au cadre la capacité non seulement de modéliser, mais aussi de comparer un nombre potentiellement illimité de théories éthiques. De plus, la compartimentation des différents types de processus permet leur analyse spécifique. Remplacer un modèle spécifique tout en maintenant les autres rend possible l'examen individualisé de ses ramifications. Sur la base de ces modèles, le *cadre d'évaluation éthique* est défini par :

$$\mathbb{F} = \langle \mathbb{A}_i, \mathbb{C}_i, \mathbb{G}_i, \mathbb{R}_i \rangle$$

Étant donné un cadre d'évaluation éthique $\mathbb{F}$ et un ensemble $\mathcal{A}$ d'actions exécutées α, nous définissons l'ensemble des actions admissibles comme suit :

$$Admissible(\mathbb{F}, \mathcal{A}) = \{\alpha \in \mathcal{A} / \mathbb{A}_i, \mathbb{C}_i, \mathbb{G}_i, \mathbb{R}_i \models admissible(\alpha)\}$$

4 Moteur d'évènement

L'Event Calculus Adapté Le moteur d'événement présenté ici correspond à l'Event Calculus complet décrit dans [27], avec les ajouts suivants. Pour répondre aux particularités de la modélisation de scénarios éthiques complexes, nous introduisons des événements automatiques en plus des actions. Ces événements automatiques se produisent lorsque toutes leurs conditions préalables, sous la forme de fluents, tiennent, sans apport de l'agent. En outre, nous

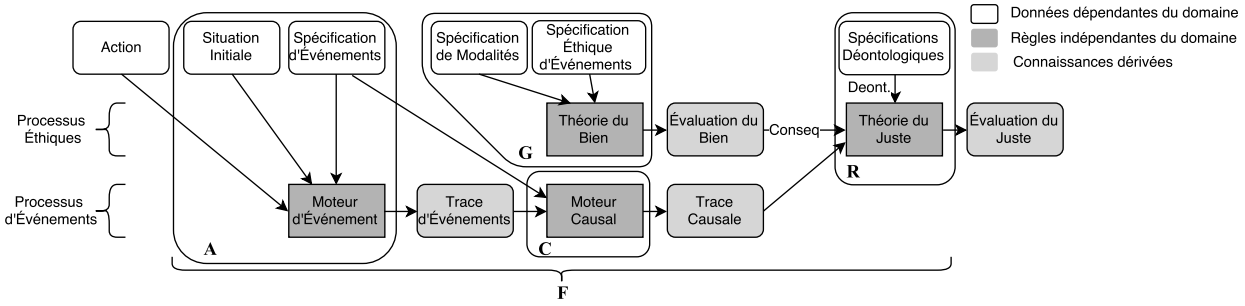

FIGURE 1 – Modèles et Modularité

faisons une distinction entre les fluents inertiels, qui restent vrais jusqu'à ce qu'ils soient terminés par une occurrence d'événement, et les fluents non inertiels qui sont vrais pour un moment seulement [24]. Enfin, nous introduisons un ensemble de simulations qui permettent à l'agent de simuler séparément et simultanément les effets de différentes actions sur le même scénario. Nous désignons l'ensemble des fonctions et des constantes comme suit : $\mathcal{S}$ est un ensemble de simulations, $\mathcal{T}$ un ensemble de moments ; $\mathcal{F}$ un ensemble de fluents, $\mathcal{A}$ un ensemble d'actions, $\mathcal{U}$ un ensemble d'événements automatiques et $\mathcal{E}$ un ensemble d'événements où $\mathcal{E} \equiv \mathcal{A} \cup \mathcal{U}$.

Axiomes d'Effets Les prédicats suivants caractérisent le comportement des fluents qui contribuent à la création d'événements. `initially(F)` indique que F est vrai initialement ; `effect(E,F)` indique que E peut causer F ; `initiates(S,E,F,T)` indique que E initie F à T dans S (et F n'est pas la négation d'un fluent) ; `terminates(S,E,F,T)` indique que E termine F à T dans S ; `clipped(S,F,T)` indique que F est stoppé à T dans S ; `nonInertial(F)` qualifie les fluents qui ne sont pas contraints par la loi d'inertie ; `holds(S,F,T)` indique que F est vrai à T dans S. Ces prédicats nous permettent d'axiomatiser les principes qui gouvernent les fluents : un fluent est vrai à T dans S si il a été initié par une occurrence d'événement à T-1 dans S ; un fluent vrai à T dans S reste vrai jusqu'à l'occurrence d'un événement qui le termine, sauf s'il est non-inertiel, auquel cas il est vrai à T seulement.

```
holds(S,F,0):-initially(F),sim(S).
initiates(S,E,F,T):-
    effect(E,F),occurs(S,E,T),not negative(S,F).
negative(S,neg(F)):-effect(E,neg(F)),sim(S).
terminates(S,E,F,T):-
    effect(E,neg(F)),occurs(S,E,T).
clipped(S,F,T):-terminates(S,E,F,T).
holds(S,F,T):-
    initiates(S,E,F,T-1),time(T).
holds(S,F,T):-
    holds(S,F,T-1),not clipped(S,F,T-1),
    not nonInertial(F),time(T).
```

Axiomes de Préconditions Les prédicats suivants caractérisent le comportement d'événements qui determinent l'état des fluents. `prec(F,E)` indique que F est une précondition de E ; `incomplete(S,E,T)` indique que E est incomplet à T dans S ; `possible(S,E,T)` indique que E est possible à T dans S ; `occurs(S,U,T)` indique que U se produit à T dans S ; `occurs(S,A,T)` indique que A se produit à T dans S. Ces prédicats nous permettent d'axiomatiser les principes qui régissent l'occurrence des événements : un événement automatique se produit à T dans S si toutes ses préconditions sont vraies à T dans S ; une action se produit à T dans S si toutes ses préconditions sont vraies et qu'un agent effectue A à T dans S.

```
incomplete(S,E,T):-
    prec(F,E),not holds(S,F,T),sim(S),time(T).
possible(S,E,T):-
    not incomplete(S,E,T),sim(S),event(E),time(T).
occurs(S,U,T):-possible(S,U,T),auto(U).
occurs(S,A,T):-
    possible(S,A,T),performs(S,D,A,T),act(A).
```

5 Moteur Causal

Axiomes de Causalité En se basant sur l'architecture de l'Event Calculus, nous définissons la causalité en termes de conséquences. Cela nous permet de générer une trace fonctionnelle et dynamique des liens causaux. Nous la définissons de la manière suivante.

Un fluent F est une *conséquence* d'un événement E si E produit F, et les deux sont vrais. Un événement E est une *conséquence* d'un fluent F si F est une précondition à E, et les deux sont vrais.

Cette définition gère la possibilité qu'il y ait plus d'une précondition pour l'occurrence de E et que F ne soit pas considéré comme une cause de E si E ne se produit pas (par exemple car d'autres préconditions n'ont pas été remplies). Pour la modéliser, nous définissons le prédicat `cons(S,E1,T,E2)`, qui indique que l'événement E2 est une conséquence de l'événement E1 qui s'est produit dans S à T. Le moment référencé est le moment auquel s'est produit le *premier* événement de la chaîne causale. Une

chaîne causale est composée d'une série de fluents et d'événements, mais le début et la fin d'une chaîne causale sont toujours des événements.

```
cons(S,E,T,F):-
    occurs(S,E,T),effect(E,F),holds(S,F,T+1).
cons(S,F,T,E):-
    occurs(S,E,T),prec(F,E),holds(S,F,T).
cons(S,E1,T1,E3):-
    cons(S,E1,T1,C2),cons(S,C2,T2,E3),
    event(E1),event(E3),T2>T1.
```

6 Théories du Bien

Dans cette section, nous présentons deux modes de définition du Bien, relatifs aux droits aux valeurs. Ces modes sont interchangeables et peuvent également être combinés. Nous présentons ensuite un modèle pour quantifier le bien une fois qu'il a été qualifié, ce qui lui permet à la fois d'être intégré dans les théories du Juste et donne aux évènements des poids significatifs. Les droits, les valeurs ou d'autres moyens de définir le bien sont appelés *modalités*.

6.1 Qualifier le Bien

6.1.1 Droits

L'*utilitarisme des droits* de Nozick postule que le Bien à maximiser consiste en la non violation des droits [25]. Un droit peut être défini comme une "*revendication justifiée que les individus et les groupes peuvent faire sur d'autres individus ou sur la société ; avoir un droit c'est être en mesure de déterminer par ses choix, ce que les autres doivent ou ne doivent pas faire*" [7]. Cette définition capture le fait qu'un droit indique à la fois un état des choses pour la personne concernée (l'exercice du droit) et une contrainte imposée aux autres (l'interdiction de violer le droit). Nous définissons les règles de sorte qu'un événement impliquant des personnes et qui viole un droit est *mal* par rapport à ce droit, et un événement qui implique des personnes mais ne viole pas un droit est *bien* par rapport à ce droit. Un événement peut être mal par rapport à un droit et bien par rapport à un autre. Cependant, aucun événement impliquant des personnes n'est neutre en ce qui concerne les droits : il respecte ou non chaque droit. Ce principe du tiers exclu est rendu explicite par l'utilisation de la négation par l'échec dans la règle. Notez que les droits sont définis par `right(M)` dans la *spécification de modalités*.

```
bad(E,X,M):-effect(E,involves(X)),
    effect(E,neg(M)),right(M).
good(E,X,M):-effect(E,involves(X)),
    not effect(E,neg(M)),right(M).
```

6.1.2 Valeurs

Une théorie fondée sur les valeurs fournit également un moyen efficace d'évaluer le mérite initial des événements. Une valeur peut être définie comme *"une conception, explicite ou implicite, distinctive d'un individu, ou caractéristique d'un groupe, de ce qui est souhaitable et qui influence la sélection des modes, des moyens et des fins d'action disponibles."* [19]. Une valeur est donc un type d'entité indépendante que peuvent assumer les actions et leurs conséquence. Les valeurs peuvent être générales, ou spécifiques à différents contextes, tels que le lieu de travail ou l'éducation des enfants. Nous définissons les règles de sorte qu'un événement qui démontre l'expression d'une valeur particulière est *bien* par rapport à cette valeur, et un événement qui démontre la négation d'une valeur est *mal* par rapport à cette valeur. Les autres événements sont considérés ni bons ni mauvais par rapport à celle-ci. Notez que les valeurs sont définies par `value(M)` dans la *spécification de modalités*.

```
good(E,X,M):-effect(E,involves(X)),
    effect(E,displays(M)),value(M).
bad(E,X,M):-effect(E,involves(X)),
    effect(E,neg(displays(M))),value(M).
```

6.2 Quantifier le Bien

Une fois déterminé le contenu du Bien, nous procédons à la quantification de ce contenu, c'est-à-dire à la 'pesée' des bonnes et mauvaises ramifications des événements. Nous définissons trois paramètres pour cela :

- Le nombre de personnes impliquées dans l'événement. Par exemple, un événement affectant cinq personnes comptera cinq fois plus qu'un événement en affectant une. Cette information est donnée dans les prédicats `good(E,X,M)` et `bad(E,X,M)` par X.
- La valeur relative des personnes impliquées dans l'événement. Par exemple, il est peut-être plus significatif de sauver des enfants plutôt que des adultes, ou de nuire à des personnes en bonne santé plutôt qu'à des patients en déclin. Ce paramètre est pris en compte par l'attribution à chaque groupe d'une valeur numérique, exprimée dans le prédicat `t_Weight(E,G,N)` où E est un événement, G son groupe cible et N le poids donné.
- L'importance de la modalité affectée par l'événement. Par exemple, être bienfaisant est peut être plus important qu'être poli, respecter le droit à la vie est peut-être plus important que respecter le droit de propriété. Ce paramètre est pris en compte par l'attribution à chaque modalité d'une valeur numérique, exprimé dans le prédicat `m_Weight(M,N)` où M est la modalité et N le poids donné.

L'attribution de poids aux modalités et aux groupes est non triviale, et la méthode proposée ici en est une parmi d'autres, qui fait acte d'introduction. Il est possible, par

exemple, de l'enrichir en prenant en compte d'autres dépendances, telles que la corrélation entre certaines modalités et personnes (par exemple, l'autonomie pourrait être essentielle pour les adultes et la sécurité pour les enfants), ou l'importance de parties affectées non humaines (les animaux, l'environnement, etc.). L'étape suivante consiste à réunir tous les poids en un seul en agrégeant par un produit les différents paramètres dans les prédicats `weightedGood(E,N,M)` et `weightedBad(E,N,M)`.

Le poids global d'un événement correspond alors à la différence entre les sommes de toutes ses bonnes et mauvaises ramifications pondérées. Plus le poids d'un événement est grand, plus il participe au Bien ; les événements qui ont des poids négatifs font plus de mal que de bien. Les poids sont donnés par le prédicat `weight(E,N)`. Ce prédicat permet l'intégration du Bien dans le Juste et participera à définir les théories du Juste. Il est à noter que les poids de cible et de modalité sont définis par `t_Weight(E,G,N)` et `m_Weight(M N)` dans la *spécification de modalités*.

```
weightedGood(E,X*N1*N2,M):-good(E,X,M),
    t_Weight(E,G,N1),m_Weight(M,N2).
weightedBad(E,X*N1*N2,M):- bad(E,X,M),
    t_Weight(E,G,N1),m_Weight(M,N2).
weight(E,N1-N3) :-
    N1=#sum[weightedGood(E,N2,M1)=N2],
    N3=#sum[weightedBad(E,N4,M2)=N4],
    number(N1;N3),event(E).
```

7 Theories du Juste

7.1 Ethiques Consequentialistes

L'éthique conséquentialiste existe sous de nombreuses formes, allant des principes simples d'action aux théories complexes pour maximiser le bien. Nous en décrivons et modélisons cinq.

7.1.1 Proscription d'Actions Purement Préjudiciables

Le premier principe conséquentialiste affirme que les actions ayant des effets purement préjudiciables sont inadmissibles. Cette règle intuitive est pertinente pour la plupart des scénarios éthiques et peut compléter d'autres théories du Juste qui ne se concentrent que sur des actions à effets complexes. Pour mettre en œuvre cette règle, nous définissons les prédicats `badCons(S,A,T)` et `goodCons(S,A,T)`, qui indiquent respectivement qu'une action A qui se produit à T dans S provoque au moins une mauvaise ou une bonne conséquence. Nous déclarons ensuite qu'une action est inadmissible si elle n'a que des mauvaises conséquences. Toute autre action qui n'a pas été démontrée inadmissible est admissible.

```
badCons(S,A,T):-act(A),cons(S,A,T,E),bad(E,X,M).
goodCons(S,A,T):-act(A),cons(S,A,T,E),good(E,X,M).
imp(pureBad,A):-
    badCons(S,A,T),not goodCons(S,A,T).
per(pureBad,A):-act(A),not imp(pureBad,A).
```

7.1.2 Principe de la Moins Mauvaise Conséquence

Également appelé *maximum minimorum*, ce principe déclare qu'une action est inadmissible si sa pire conséquence est pire que la pire conséquence de toute autre action possible. Ce principe est particulièrement pertinent pour la prise de décision sous incertitude, où, sous 'l'hypothèse de la malchance', chaque action possible produirait sa pire conséquence. Ainsi l'agent ferait mieux de choisir l'alternative ayant le moins mauvais mauvais résultat [22]. Pour formaliser cette règle, nous déterminons d'abord une hiérarchie entre les conséquences des actions, afin d'ensuite indiquer la pire. Le prédicat `worse(E1,E2)` indique que la conséquence E1 d'une action est pire que la conséquence E2 de la même ou d'une autre action. Les prédicats `notWorstCons(S,A,T,E)` et `worstCons(S,A,T,E)` déterminent alors la limite basse d'un ordre partiel déterminé par le prédicat `worst`. Enfin, nous déclarons qu'une action A1 est inadmissible si sa pire conséquence E1 est pire que la pire conséquence E2 de toute autre action A2. Toutes les autres actions sont admissibles.

```
worse(E1,E2):-
    cons(S1,A1,T1,E1),cons(S2,A2,T2,E2),
    weight(E1,N1),weight(E2,N2), N1<N2.
notWorstCons(S,A,T,E1):-
    act(A),cons(S,A,T,E1),cons(S,A,T,E2),
    worse(E2,E1),not worse(E1,E2).
worstCons(S,A,T,E):-act(A),event(E),
    cons(S,A,T,E),not notWorstCons(S,A,T,E).
imp(leastBad,A1):-worstCons(S1,A1,T1,E1),
    worstCons(S2,A2,T2,E2),
    worse(E1,E2),A1!=A2.
per(leastBad,A):-act(A),not imp(leastBad,A).
```

7.1.3 Principe d'Analyse Coût-Avantage

Ce principle indique qu'une action est admissible si elle est globalement bénéfique, c'est-à-dire si ses bonnes conséquences l'emportent les mauvaises. Nous utilisons les prédicats `weightCons(S,A,T,E,N)`, qui détermine le poids N des conséquences individuelles E d'une action A qui s'est produite à T dans S, et `weightAct(A,N2)`, qui concatène ces poids [2], pour indiquer que A est inadmissible si ce poids total N2 est négatif, et admissible sinon.

```
weightCons(S,A,T,E,N):-
    act(A),cons(S,A,T,E),weight(E,N).
weightAct(A,N2):-act(A),number(N2),
    N2=#sum[weightCons(S,A,T,E,N1)=N1].
imp(benCosts,A):-weightAct(A,N2),number(N2), N2<0.
per(benCosts,A):-act(A),not imp(benCosts,A).
```

2. sans spécifier de situation car une seule action est réalisée dans chaque situation.

7.1.4 Utilitarianisme de l'Acte

> *"C'est le plus grand bonheur du plus grand nombre qui est la mesure du bien et du mal."* J. Bentham, 1776 [8].

L'utilitarisme de l'acte exige que l'on évalue une action directement selon le *principe d'utilité*, qui stipule que l'action moralement correcte est celle qui a les meilleures conséquences globales pour le bien-être ou l'utilité de la majorité des parties concernées [8]. Une action est donc admissible si, compte tenu de toutes les autres actions disponibles, elle a les meilleures conséquences dans l'ensemble. En utilisant le prédicat `weightAct` défini ci-dessus, nous déterminons un ordre de préférence entre les actions dans le domaine et déclarons qu'une action A1 est inadmissible s'il existe une autre action A2 dont le poids est supérieur. Toute autre action est admissible.

```
imp(actUti,A1):-
    weightAct(A1,N1),weightAct(A2,N2),N1<N2.
per(actUti,A):-act(A),not imp(actUti,A).
```

7.1.5 Utilitarisme de la Règle

> *"Chaque acte, dans la vie morale, tombe sous une règle; et nous devons juger la moralité ou l'immoralité de l'acte, non par ses conséquences, mais par les conséquences de son universalisation -c'est-à-dire par les conséquences de l'adoption de la règle sous laquelle cet acte tombe." J. Hospers, 1975 [28]*

L'utilitarisme de la règle évalue une action en deux temps. La première étape consiste à évaluer les règles morales sur la base du principe d'utilité : il s'agit de déterminer si une règle ou un ensemble de règles morales engendrera les meilleures conséquences, supposant que tout ou la plupart des agents s'y plient. Dans la vie quotidienne, de telles règles peuvent inclure 'Ne pas voler', ou 'Gardez ses promesses'. La deuxième étape consiste à évaluer les actions individuelles relativement à ce qui a été justifié au cours de la première étape. Une action n'est admissible que si la règle à laquelle elle appartient respecte le principe d'utilité, outre son propre respect du principe. Par exemple, si 'Ne pas voler' est une règle adoptée, le vol sera toujours inadmissible, même dans l'instance ou un vol produirait la plus grande utilité (par exemple, en alimentant un affamé). Le prédicat `ruleCount(R,N)` regroupe tous les poids N des actions qui appartiennent à une règle R ; le prédicat `weightRule(R,N)` les concatène. Nous déclarons alors qu'une action A est inadmissible si elle est une instance d'une règle R globalement nuisible, c'est-à-dire dont les mauvaises conséquences l'emportent sur les bonnes, considérant toutes ses instanciation. Toute autre action est admissible. Il est à noter que les instances de règles et les règles sont définies par `rule(R)` et `instance(A,R)` dans la *spécification de modalités*.

```
ruleCount(R,N):-
    rule(R),instance(A,R),weightAct(A,N).
weightRule(R,N):-
    rule(R),number(N),N=#sum[ruleCount(R,N1)=N1].
imp(ruleUti,A):-
    act(A),instance(A,R),weightRule(R,N),N<0.
per(ruleUti,A):-act(A),not imp(ruleUti,A).
```

7.2 Éthiques Déontologiques

Dans cette section, nous présentons trois doctrines déontologiques. Deux d'entre elles sont purement déontologiques, les codes de conduite et l'éthique kantienne. La doctrine du double effet comporte des contraintes conséquentialistes. Il est à noter que notre traduction logique de ces théories constitue une possibilité parmi d'autres.

7.2.1 Codes de Conduite

> *"J'apporterai mon aide à mes confrères ainsi qu'à leurs familles dans l'adversité. Que les hommes et mes confrères m'accordent leur estime si je suis fidèle à mes promesses; que je sois déshonoré et méprisé si j'y manque." Serment d'Hippocrate, [12]*

Un code de conduite est un ensemble de règles qui décrit les obligations, les interdictions ou les responsabilités d'un individu, d'un groupe ou d'une organisation. Il spécifie les principes qui guident la prise de décision ou les procédures de ceux qui sont contraints par le code. Les codes de conduite varient dans leur portée et leur nature, allant des codes déontologiques professionnels aux commandements religieux. Le comportement et la moralité sont typiquement déterminés par un corps global, tel qu'une entreprise, un état ou un dieu. Nous illustrons ici ce type de contrainte en modélisant une règle commune qu'est l'interdiction de tuer. Une telle règle se trouve par exemple dans la Déclaration de Genève de l'Association Médicale Mondiale sous la forme de la déclaration «Je garderai le respect absolu de la vie humaine» [6], ou dans le Décalogue comme commandement 'Tu ne tueras point'(Exode 20 : 1-21). Nous modélisons une règle de ce type en déclarant qu'une action est inadmissible dans la mesure où elle provoque ou consiste en ce qui est interdit, ici, tuer.

```
imp(conduct,A):-act(A),cons(S,A,T,kill(N,G)).
per(conduct,A):-act(A),not imp(conduct,A).
```

7.2.2 La Formule de la Fin en Elle-Même

> *"Agis de telle sorte que tu traites l'humanité aussi bien dans ta personne que dans la personne de tout autre toujours en même temps comme une*

fin, et jamais simplement comme un moyen." I. Kant, 1785 [18]

Cette formule est un élément d'éthique kantienne qui met l'accent sur la valeur intrinsèque de la vie humaine. C'est un impératif moral qui interdit d'utiliser les personnes comme moyen pour d'autres fins, les personnes étant des fins en elles-mêmes en vertu de leur nature d'êtres rationnels [18]. La formule contraste la valeur intrinsèque, qui est persistante et souveraine, avec la valeur instrumentale, qui dépend de ce qu'elle produit. Nous présentons le prédicat `aim(A,E)` qui indique que le but de l'action A est de provoquer l'événement E et utilisons le fluent `involves(X)` pour indiquer qu'au moins une personne est impliquée dans E. Nous déclarons alors qu'une action A est inadmissible si elle provoque un événement E qui implique au moins une personne, mais où E n'est pas un but de A. Toute autre action est admissible. Les buts sont définis par `aim(A,E)` dans la *spécification déontologique*.

```
imp(kant,A):-act(A),cons(S,A,T,E),
    effect(E,involves(X)),not aim(A,E).
per(kant,A):-act(A),not imp(kant,A).
```

7.2.3 La Doctrine du Double Effet

'Rien n'empêche un acte d'avoir deux effets, dont un seul est voulu, tandis que l'autre est à côté de l'intention." T. Aquinas, 1485 [4]

La doctrine du double effet est un ensemble de critères éthiques utilisés pour évaluer la permissibilité éthique d'une action qui a à la fois de bonnes et de mauvaises conséquences [15]. Elle dicte qu'une personne peut licitement exécuter une action en sachant qu'elle aura des bons et mauvais effets, à condition que : 1) L'action elle-même soit bonne ou moralement neutre ; 2) Le mauvais effet ne soit pas directement voulu ; 3) Le bon effet résulte de l'acte et non du mauvais effet ; 4) Le bon effet soit plus important ou égal au mauvais effet [23]. `imp(dde1,A)` proscrit une action si elle est intrinsèquement mauvaise, correspondant à la condition 1. `imp(dde2,A)` proscrit une action si elle provoque un mauvais effet qui conduit à un bon effet. Cette règle correspond aux conditions 2 et 3, car nous estimons que l'utilisation d'un événement comme moyen pour arriver à un autre événement équivaut à vouloir le premier événement. `imp(dde3, A)` proscrit une action si son effet global est mauvais. Cela correspond à la condition 4 qui équivaut au Principe d'Analyse Coût-Avantage. Toutes les autres actions sont permises par la doctrine.

```
imp(dde1,A):-act(A),bad(A,X,M).
imp(dde2,A):-
    act(A),cons(S,A,T,E1),cons(S,E1,T2,E2),
    bad(E1,X1,M1), good(E2,X2,M2).
imp(dde3,A):-imp(benefitsCosts,A).
per(dde,A):-act(A),not imp(dde1,A),
    not imp(dde2,A),not imp(dde3,A).
```

7.2.4 Discussion

La modélisation de ces théories souligne certaines de leurs caractéristiques. D'abord, nous distinguons deux types de théories du Juste : celles qui évaluent chaque action par rapport aux autres actions possibles, et celles qui évaluent chaque action de manière indépendante. Les principes *relatifs* comparent les actions et font un choix unique. Le *principe de la moins mauvaise conséquence*, l'*utilitarisme de l'acte* et l'*utilitarisme de la règle* sont de ce genre. Cependant, l'*utilitarisme de la règle* est particulier en ce sens que la permissibilité de *toutes* les actions est déterminée selon l'impact de chacune individuellement. Ainsi, toutes ou aucune des actions considérées sous une règle sont admissibles. À l'inverse, les principes *indépendants* évaluent chaque action en elle-même sans être affectés par les options disponibles. Ainsi, ils produisent des ensembles d'actions admissibles ou inadmissibles. Toutes les autres théories du Juste présentées ici sont de ce genre. Il est important de noter que différentes théories du Juste peuvent, et parfois doivent, se compléter. Par exemple, la *doctrine du double effet* ne dit rien sur les action dont les effets sont purement mauvais et peut être utilement complétée par un principe conséquentialiste pour y remédier.

8 Preuve de Concept

Dans cette section, nous illustrons la manière dont chaque contrainte éthique décrite ci-dessus gère un dilemme éthique à l'aide d'un exemple mono-agent de prise de décision. C'est une exemple jouet qui n'a pas vocation à être réaliste, mais vise à montrer la diversité des évaluations éthiques possibles. Le code source complet est téléchargeable sur un service cloud [3].

Un Dilemme Médical Un médecin (l'agent autonome) est en possession de trois différents traitements expérimentaux pour une maladie grave et handicapante. Chaque traitement a un taux de réussite différent.

- Pour 100 patients qui essaient le traitement Alpha, 15 sont guéris, 20 perdent leur vie et 65 restent inchangés.

- Pour 100 patients qui tentent le traitement Bêta, 30 sont guéris, 25 perdent leur vie et 45 restent inchangés.

- Pour 100 patients qui essaient le traitement Gamma, 50 sont guéris, 30 perdent leur vie et 20 restent inchangés. Cependant, sur les 50 patients guéris, 30 ne sont entièrement guéris que parce qu'ils ont également reçu une transplantation d'organe provenant de chacun des 30 morts. Sans la transplantation, ils auraient perdu la vie.

Le gain en termes de vies sauvées (c'est-à-dire les patients guéris moins les patients tués) pour chaque traitement est donc : **Alpha -5** ; **Beta 5** ; **Gamma 20**. Afin de choisir quel traitement est admissible, nous considérons

3. https ://gitlab.lip6.fr/ganascia/AAMAS_2017

que le médecin simule séparément le fait de donner chaque traitement à un groupe de 100 personnes. Il administre le traitement à T=0 dans trois simulations respectives. Le cas est représenté de la manière suivante.

```
act(give(Z)):-treatment(Z).
prec(exist(Z),give(Z)):-act(give(Z)).
performs(sim1,doctor,give(alpha),0).
performs(sim2,doctor,give(beta),0).
performs(sim3,doctor,give(gamma),0).
initially(exist(Z)):-treatment(Z).
auto(cure(X,G);kill(X,G);null(X,G))
    :-people(X), group(G).
testgroup(group(alpha;beta;gamma)).
treatment(alpha;beta;gamma).
effect(give(alpha),posImpact(15,group(alpha))).
effect(give(alpha),negImpact(20,group(alpha))).
effect(give(alpha),noImpact(65,group(alpha))).
effect(give(beta),posImpact(30,group(beta))).
effect(give(beta),negImpact(25,group(beta))).
effect(give(beta),noImpact(45,group(beta))).
effect(give(gamma),posImpact(20,group(gamma))).
effect(give(gamma),negImpact(30,group(gamma))).
effect(give(gamma),noImpact(20,group(gamma))).
effect(kill(X,group(gamma)),
    posImpact(X,group(gamma))):- people(X).
prec(posImpact(X,G),cure(X,G)):-auto(cure(X,G)).
prec(negImpact(X,G),kill(X,G)):-auto(kill(X,G)).
prec(noImpact(X,G),null(X,G)):-auto(null(X,G)).
effect(cure(X,G),involves(X)):-auto(cure(X,G)),X>0.
effect(null(X,G),involves(X)):-auto(null(X,G)),X>0.
effect(kill(X,G),involves(X)):-auto(kill(X,G)),X>0.
nonInertial(posImpact(X,G);negImpact(X,G);
noImpact(X,G);involves(X)):-people(X),testgroup(G).
```

Le docteur considère en outre que venir en aide constitue le Bien, que guérir est venir en aide, tuer est l'opposé de venir en aide et laisser la maladie inchangée est n'avoir aucun impact. Il considère également que la valeur *venir en aide* (`helpfullness`) a un poids de 1 (ceci est ici trivial car il n'y a qu'une seule modalité) et que la vie de tous les patients est équivalente. Il pense aussi que donner chacun de ces traitements pourrait être généralisé comme la règle 'donner des remèdes incertains'. Enfin, son objectif en donnant des traitements est de guérir.

```
m_Weight(M,1):-modality(M).
t_Weight(E,G,1):-
    testgroup(G),effect(E,involves(X)).
modality(M):-value(M).
value(helpfulness).
effect(cure(X,G),displays(helpfulness)):-
    auto(cure(X,G)).
effect(kill(X,G),neg(displays(helpfulness))):-
    auto(kill(X,G)).
rule(uncertainCures).
instance(give(alpha;beta;gamma),uncertainCures).
aim(give(Z),cure(X,group(Z))):-
    treatment(Z),people(X).
```

TABLE 1 – Evaluations Basées sur la Valeur 'helpfulness'

	Alpha	**Beta**	**Gamma**
pureBad	Admis	Admis	Admis
leastBad	Admis	Inadmis	Inadmis
benCosts	Inadmis	Admis	Admis
actUti	Inadmis	Inadmis	Admis
ruleUti	Admis	Admis	Admis
conduct	Inadmis	Inadmis	Inadmis
kant	Inadmis	Inadmis	Inadmis
dde	Inadmis	Admis	Inadmis

TABLE 2 – Evaluations Basées sur le Droit 'life'

	Alpha	**Beta**	**Gamma**
pureBad	Admis	Admis	Admis
leastBad	Admis	Inadmis	Inadmis
benCosts	Admis	Admis	Admis
actUti	Admis	Inadmis	Inadmis
ruleUti	Admis	Admis	Admis
conduct	Inadmis	Inadmis	Inadmis
kant	Inadmis	Inadmis	Inadmis
dde	Admis	Admis	Inadmis

Les résultats de l'évaluation éthique sont résumés dans le tableau 1. Pour montrer comment le remplacement d'un module par un autre peut changer le processus d'évaluation, nous modélisons également un cas dans lequel le docteur fonde sa théorie du Bien non pas sur une *valeur* mais sur le respect du *droit* à la vie. Cette modalité est définie comme suit, et les résultats résumés dans le tableau 2.

```
modality(M):-right(M).
right(life).
effect(kill(X,G),neg(life)):-auto(kill(X,G)).
```

9 Conclusion

Le cadre présenté ici adapte et s'appuie sur l'Event Calculus pour permettre la modélisation de théories éthiques et de scénarios dans lesquels les appliquer. Défini en programmation logique, il présente une méthode et une implémentation de cette méthode. L'accent est mis sur la hiérarchie et la représentation explicite des processus de raisonnement qui déterminent la prise de décision éthique. Celles-ci permettent d'engendrer des règles avec un fort potentiel expressif qui confèrent aux agents la capacité de décider et d'expliquer leurs décisions, mais aussi de raisonner sur les actions d'autres agents. En outre, la confrontation de théories éthiques avec les contraintes logiques des langages de programmation éclaire ces théories, clarifiant leurs concepts, les relations qui les lient et les ambiguïtés potentielles qu'elles peuvent contenir. Nous envi-

sageons un certain nombre d'avenues futures pour développer le cadre actuel. Tout d'abord, nous cherchons à modéliser l'intention, qui n'est pour l'instant traitée qu'implicitement, et de modéliser les désirs des agents. Cela permettra aux agents de gérer des scénarios plus complexes et plus réalistes. En outre, nous avons l'intention de permettre la formulation de plans éthiques d'actions dans lesquels plus d'une action peut être évaluée dans une simulation, en travaillant vers un véritable domaine de planification. Enfin, nous visons à intégrer le cadre dans un système multi-agents, afin d'exploiter plus pleinement son potentiel pour faciliter la coopération ou l'intelligence collective.

Remerciements

Les auteurs remercient l'Agence Nationale de la Recherche (ANR) pour sa contribution financière sous la référence ANR-13-CORD-0006.

Références

[1] Alexander, L et M Moore: *Deontological Ethics*. Dans Zalta, Edward N. (rédacteur) : *The Stanford Encyclopedia of Philosophy*. 2016.

[2] Anderson, M et S Anderson: *Machine ethics*. Cambridge University Press, 2011.

[3] Anderson, M, S Anderson et C Armen: *MedEthEx : a prototype medical ethics advisor*. 2006.

[4] Aquinas, T: *Summa theologica*. Xist Publishing, 2015.

[5] Arkin, R: *Governing lethal behavior in autonomous robots*. CRC Press, 2009.

[6] Association, World Medical *et al.*: *WMA declaration of Geneva*. International Journal of Person Centered Medicine, 4(3), 2015.

[7] Beauchamp, T et J Childress: *Principles of Biomedical Ethics*. Principles of Biomedical Ethics. Oxford University Press, 2001, ISBN 9780195143317.

[8] Bentham, J: *A fragment on government*. The Lawbook Exchange, Ltd., 2001.

[9] Berreby, F, G Bourgne et J G Ganascia: *Modelling Moral Reasoning and Ethical Responsibility with Logic Programming*. Dans *Logic for Programming, Artificial Intelligence, and Reasoning*, pages 532–548. Springer, 2015.

[10] Bringsjord, S et J Taylor: *The divine-command approach to robot ethics*. Robot Ethics : The Ethical and Social Implications of Robotics', MIT Press, Cambridge, MA, pages 85–108, 2012.

[11] Cointe, N, G Bonnet et O Boissier: *Ethical Judgment of Agents' Behaviors in Multi-Agent Systems*. Dans *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 1106–1114, 2016.

[12] Cos, Hippocrates of: *The Oath*. Loeb Classical Library, 1923.

[13] Dennis, L, M Fisher, M Slavkovik et M Webster: *Formal verification of ethical choices in autonomous systems*. Robotics and Autonomous Systems, 77 :1–14, 2016.

[14] Dennis, LA, M Fisher et A Winfield: *Towards verifiably ethical robot behaviour*. arXiv preprint arXiv :1504.03592, 2015.

[15] Foot, P: *The problem of abortion and the doctrine of the double effect*. Applied Ethics : Critical Concepts in Philosophy, 2 :187, 2002.

[16] Ganascia, J G: *Non-monotonic resolution of conflicts for ethical reasoning*. Dans *A Construction Manual for Robots' Ethical Systems*, pages 101–118. Springer, 2015.

[17] Horty, J: *Nonmonotonic foundations for deontic logic*. Dans *Defeasible deontic logic*. Springer, 1997.

[18] Kant, I: *Groundwork of the Metaphysic of Morals, trans. HJ Paton*. New York : Harper & Row, 1964.

[19] Kluckhohn, C: *Values and value-orientations in the theory of action : An exploration in definition and classification*. 1951.

[20] Kowalski, R: *Computational logic and human thinking : how to be artificially intelligent*. Cambridge University Press, 2011.

[21] Lifschitz, V: *What Is Answer Set Programming ?*. Dans *AAAI*, tome 8, pages 1594–1597, 2008.

[22] Luce, D et H Raffia: *Games and decisions. Mineola, NY*, 1985.

[23] Mangan, J T: *Historical Analysis of the Principle of Double Effect, An*. Theological Studies, 10, 1949.

[24] Miller, R et M Shanahan: *Some alternative formulations of the event calculus*. Dans *Computational logic : logic programming and beyond*. Springer, 2002.

[25] Nozick, R: *Anarchy, state, and utopia*, 1974.

[26] Pereira, LM et A Saptawijaya: *Modelling morality with prospective logic*. Dans *Progress in Artificial Intelligence*, pages 99–111. Springer, 2007.

[27] Shanahan, M: *The event calculus explained*. Dans *Artificial intelligence today*. Springer, 1999.

[28] Struhl, KJ et PS Rothenberg: *Ethics in perspective : a reader*. Random House, 1975.

[29] Tufiş, M et J G Ganascia: *Grafting norms onto the BDI agent model*. Dans *A Construction Manual for Robots' Ethical Systems*. Springer, 2015.