INCLUSION EXCLUE : LE CODE EST UN CONTRAT LÉONIN

Enquête sur la valeur technique et juridique du protocole robots.txt

Guillaume Sire

La Découverte | « Réseaux »

2015/1 n° 189 | pages 187 à 214

ISSN 0751-7971
ISBN 9782707185419

Article disponible en ligne à l'adresse :

https://www.cairn.info/revue-reseaux-2015-1-page-187.htm

Pour citer cet article :

Guillaume Sire, « Inclusion exclue : le code est un contrat léonin. Enquête sur la valeur technique et juridique du protocole robots.txt », *Réseaux* 2015/1 (n° 189), p. 187-214.
DOI 10.3917/res.189.0187

INCLUSION EXCLUE : LE CODE EST UN CONTRAT LÉONIN

Enquête sur la valeur technique et juridique du protocole robots.txt

Guillaume SIRE

DOI: 10.3917/res.189.0187

On a beau lire de nombreux commentaires au sujet du prétendu pouvoir des langages de programmation et de leur supposée valeur de loi, on trouve moins souvent des lignes de code citées dans les revues de sciences sociales. Rares sont les articles qui analysent les variables, les chaînes de caractères, les fonctions algorithmiques et les balises métatextuelles. Pourtant, le code informatique est un langage qui a ses structures et son vocabulaire (Cox *et al.*, 2001 ; Cayley, 2002 ; Glazier, 2006). Dès lors, il convient d'étudier le rôle joué par ce langage dans l'espace social, car l'acte de coder, comme l'acte de parler ou d'écrire, « pose un *contrat avec l'autre* dans un réseau de places et de relations » (De Certeau, 1980, p. 38). C'est ce *contrat* que la célèbre phrase de Lawrence Lessig – « le code, c'est la loi » (Lessig, 2000, 2006) – invite à étudier. Un tel aphorisme ne peut en effet pas tant être considéré comme un constat que comme une mise en garde : celui qui exclut *a priori* le code de l'étude d'une chaîne de médiations où il intervient risque de passer à côté des dynamiques sociales propres au phénomène de « computation » (Cramer, 2005 ; Fuller et Marino, 2013). Mark Marino soutient par conséquent que les sciences sociales auraient beaucoup à apprendre et à apporter en se penchant sur le code : « *Tandis que les chercheurs en informatique théorisent la façon dont le code peut être le plus utile, explique-t-il, les sciences sociales peuvent aider à comprendre la signification du code* » (Marino, 2006). Autrement dit, tandis que les informaticiens se posent la question « Qu'est-ce que coder peut faire ? », il s'agit pour les sciences sociales de se demander : « Qu'est-ce que coder *veut dire*, et qu'est-ce que cela implique socialement ? »

Même si c'est loin d'être courant, certains anthropologues et sociologues, ainsi que certains philosophes, se sont tout de même intéressés au rôle social du code (Manovich 2001, 2013 ; Fuller 2003, 2008 ; Berry, 2011). Se réclamant en général d'un champ nommé « *software studies* », ils stipulent dans leurs travaux que le code renvoie à l'environnement au sein duquel il est utilisé, de la même manière qu'une langue renvoie à la société qui la parle et à la culture qu'elle nourrit. Étudier la « politique des algorithmes »[1] reviendrait

1. Cf. *Réseaux*, n° 177.

selon cette approche à étudier les acteurs qui se saisissent du code pour agir et faire agir, en même temps qu'on étudierait la réalité dans laquelle le code s'inscrit, par laquelle il agit, et qu'éventuellement il transforme. Les « *critical code studies* » diffèrent légèrement des *software studies* en cela qu'elles préconisent d'étudier, outre l'environnement, le code lui-même (Marino, 2006).

Internet fonctionne grâce à un écheveau de protocoles. Chacun de ces protocoles est constitué d'un code et d'une syntaxe, c'est-à-dire d'un ensemble de règles prévoyant que tel ou tel usage du code induira tel ou tel effet. Si un acteur se réfère à un protocole en particulier, il doit veiller à utiliser correctement sa syntaxe, d'une part, et, d'autre part, s'assurer que les concepteurs des machines à qui le code est adressé aient fait en sorte de respecter à la lettre le protocole en question. Le codeur doit en outre composer avec « *les contraintes imposées par le fait que tout langage de programmation est un langage formel qui ne permet pas de formulations ambiguës et n'accepte qu'une syntaxe parfaite* » (Rieder, 2006, p. 243).

« *Les protocoles de l'Internet sont l'Internet lui-même, explique Lara DeNardis. Ils sont les règles fondamentales qui permettent aux dispositifs d'échanger des informations. Ces règles requièrent qu'un accord soit trouvé entre les acteurs impliqués* » (DeNardis, 2014, p. 66). Les acteurs de l'Internet doivent en effet réussir à se mettre d'accord pour standardiser des protocoles qui permettront de coordonner leurs actions. Les parties prenantes se réunissent en général sous forme de groupes de travail et discutent à partir d'une première version d'un protocole, exprimée par l'un d'eux sous forme de *Request For Comments* (RFC), débattue et amendée jusqu'à ce que les modalités d'exécution soient stabilisées. Ces discussions ont lieu sous la houlette d'instances de standardisation, notamment l'*Internet Engineering Task Force* (IETF) pour l'Internet et le *World Wide Web Consortium* (W3C) pour ce qui a trait au web en particulier. Dans les groupes de discussion chapeautés par ces instances, les participants n'exercent pas tous la même influence. Cependant n'importe qui peut théoriquement participer : entreprises, associations, individus. Les participants se livrent à ce que l'on pourrait nommer une « sociologie implicite de la communication » : dès lors que chacun d'eux prévoit une utilité sociale au protocole ou à l'amendement qu'il propose, c'est qu'il pense *quelque chose* à propos de la société à laquelle il destine ce protocole ou cet amendement. Des négociations ouvertes ont lieu, plus ou moins virulentes, durant lesquelles chacun défend ses intérêts et sa « vision » en composant avec d'autres intérêts et d'autres « visions ». Des alliances sont passées, des

disputes éclatent, des compromis sont trouvés. Le protocole résulte au final de la rencontre de ce que la sociologie des techniques nomme le « programme d'action » de chacun des participants, c'est-à-dire d'une « série d'objectifs, d'états et d'intentions qu'un agent peut parcourir » (Latour 1999), ainsi que des détours nécessaires à la conciliation de ces programmes d'action. En nous apprenant ce que les acteurs concernés par un protocole auraient voulu que celui-ci devienne, l'étude des négociations permet de sonder les rapports de forces des parties prenantes et d'éclairer ce que le protocole est effectivement devenu, et comment, à l'issue des négociations.

Nous nous intéressons ici à la manière dont le protocole d'exclusion « robots.txt » permet à l'éditeur d'un site web de réguler l'activité des moteurs de recherche sur ses pages. Ce protocole, basé sur l'*American Standard Code for Information Interchange* (ASCII), a pour but principal de permettre à un éditeur d'exclure certains de ses documents du champ d'action des agents logiciels appelés « *crawlers* » utilisés par les moteurs de recherche pour prendre connaissance des documents. La syntaxe ASCII propre au protocole ne peut avoir d'effet que parce que l'éditeur et le concepteur du moteur sont tous les deux au fait des modalités qui prévalent à son usage et des effets que celui-ci est censé produire. Ce n'est pas pour autant qu'ils ont accepté de bon cœur ces modalités. La syntaxe usuelle du protocole d'exclusion est en effet l'objet d'une controverse, certains éditeurs considérant qu'elle ne suffit pas à garantir leurs droits. Ainsi, ce protocole, comme c'est souvent le cas sur Internet, sert de « lieu de conflit entre différents intérêts sociaux et économiques » (DeNardis, 2014, p. 64). Ci-après, nous décrirons la solution alternative qui fut proposée par des éditeurs de contenus à la fin des années 2000, mais fut largement rejetée. Comparer l'alternative rejetée au protocole établi nous aidera, conformément à ce que suggère Laura DeNardis, « *à mettre à jour les valeurs et les intérêts en jeu dans le développement et la sélection [des protocoles de l'Internet]* » (*ibid*.).

Nous amorcerons notre argument en évoquant les rapports conflictuels que les éditeurs ont avec les moteurs de recherche au sujet du droit d'auteur, et en particulier avec Google qui se trouve en position dominante dans ce secteur (68 % des parts de marché dans le monde[2] et 94 % en France[3]). Ce point d'entrée nous conduira à décrire le protocole d'exclusion et sa syntaxe, après

2. Net Market Share, *Desktop Search Engine Market Share*, août 2014.
3. AT Internet, *Baromètre des moteurs de recherche*, août 2014.

quoi nous présenterons le projet « *Automated Access Content Protocol* » (ACAP), soutenu en 2009 par 1250 éditeurs, dans le but de remplacer le protocole d'exclusion. Nous comparerons les principales commandes de l'ACAP à celles du protocole d'exclusion pour voir ce qu'elles nous enseignent sur ce que *fait* techniquement le code ASCII et sur ce qu'il *veut dire* juridiquement. Enfin, nous expliquerons comment ACAP a échoué et en tirerons des conclusions quant au rapport de forces de ceux qui produisent le code et de ceux qui conçoivent les machines destinées à l'exécuter.

GOOGLE, LES ÉDITEURS ET LE PROTOCOLE D'EXCLUSION

Le rapport des éditeurs de contenus avec les propriétaires de moteurs de recherche, Google en tête, est caractéristique d'une situation mi-coopérative, mi-compétitive, dite « de *coopétition* » : le moteur de recherche coopère avec les éditeurs en leur envoyant du trafic tandis que la firme à laquelle il appartient les concurrence sur le marché de la publicité (Smyrnaios et Rebillard, 2011). Aux yeux de certains éditeurs, cette concurrence est déloyale. Ceux-là se plaignent du fait que la firme Google (sur laquelle nous concentrerons l'essentiel de notre argument étant donné la position dominante de son moteur de recherche) génère des revenus grâce à leurs contenus en y juxtaposant de la publicité, et qu'il faudrait par conséquent les rémunérer au nom du droit d'auteur. Google leur répond qu'il s'agit d'un échange de contenu contre visibilité et qu'il n'est pas question de leur verser quoi que ce soit. En outre, à chaque fois qu'un éditeur reproche à Google de ne pas respecter le droit d'auteur, les porte-parole de la firme avancent un même argument : puisqu'il est possible d'exclure la totalité ou une partie de son site de l'index du moteur grâce au protocole d'exclusion robots.txt, c'est, dès lors que l'éditeur ne le fait pas, qu'il accepte que ses documents soient référencés selon les modalités prévues par les concepteurs du moteur. Ainsi, lorsque plusieurs éditeurs de contenus ont signé une déclaration commune pour prétendre qu'ils « *ne souhaitaient plus qu'on les force à abandonner la possibilité de jouir du droit de propriété qu'ils avaient vis-à-vis de leurs contenus sans leur demander de permission* », Google leur a répondu en faisant référence au robots.txt : « *Si vous ne voulez pas apparaître dans les résultats de Google, cela ne requiert qu'une ou deux lignes de code* » (Cohen, 2009). Rejoignant l'argument de la firme, l'expert Dan Sullivan suggère aux éditeurs mécontents de l'action de Google de mettre un « préservatif robots.txt » pour protéger leurs contenus (Sullivan, 2009a). Sullivan adresse son conseil en particulier à Rupert Murdoch, propriétaire de

nombreux titres de presse *via* sa firme News Corp (*Wall Street Journal, New York Post, The Daily, The Times, The Sun…*) accusant Google de se livrer à une concurrence déloyale sur le marché de la publicité où la firme profiterait selon lui de la valeur ajoutée apportée par les contenus de *News Corp* à l'index de son moteur, sans redistribuer à *News Corp* une partie des revenus dégagés. C'est pour cette même raison qu'en France, Josh Cohen, responsable de Google Actualités, fut très mal accueilli lors de son intervention aux *États généraux de la presse* de décembre 2008 (Sire, 2013). Malgré ces différends, Google a continué de considérer que l'absence de recours au robots.txt sur les sites des éditeurs avait valeur d'« accord tacite ». Pour autant, la justice de plusieurs pays ne l'a pas entendu de cette oreille. Par exemple, dans le cas de la controverse qui a opposé Google aux éditeurs de presse belge lui reprochant de violer leur droit d'auteur (Sookman, 2011 ; Strowel, 2011), tandis que Google brandissait son argument habituel, la justice belge a statué que le droit d'auteur n'était pas un droit de ne pas être repris par un tiers, mais un droit interdisant à un tiers de reprendre un contenu sans l'autorisation *a priori* du propriétaire et de l'auteur du contenu en question (Van Asbroeck et Cock, 2007, p. 466). De telles décisions de justice n'ont pas suffi à changer la politique de Google qui continue de préconiser aux éditeurs mécontents le recours au robots.txt. La controverse est par conséquent toujours aussi virulente : en septembre 2014, Rupert Murdoch a accusé Google de « manger vivant son hamster » alors que le CEO de *News Corp*, Robert Thomson, venait de rendre publique une lettre adressée au vice-président de la Commission européenne, Joaquín Almunia, pour accuser Google de piraterie et de cynisme vis-à-vis de la règle de droit (Rushe, 2014).

Le nœud principal de cette controverse vient du fait que les éditeurs ne veulent pas désindexer leurs contenus du moteur, mais négocier les modalités de cette indexation. Google s'y oppose, arguant que ces modalités ne sont pas négociables : soit l'éditeur exclut son site du périmètre d'action du moteur grâce au protocole robots.txt, soit il accepte que les *crawlers* de Google visitent son contenu. Certains éditeurs ont décidé par conséquent de modifier la syntaxe du robots.txt pour que le protocole d'exclusion soit remplacé par un protocole d'inclusion au sein duquel il ne s'agirait plus d'inclure dans l'index du moteur les pages qui n'en sont pas explicitement exclues, mais au contraire d'exclure les pages qui n'y ont pas été explicitement incluses. Nous décrirons ci-après le fonctionnement du robots.txt, avant d'en venir à l'analyse de cette tentative de changement de protocole, laquelle nous permettra d'expliquer quelles conditions doivent être réunies pour qu'un protocole ayant recours au

ASCII puisse être stabilisé, et de montrer pourquoi c'est aux concepteurs de machines à traduire le code, Google en tête, et non à ceux qui écrivent le code, qu'appartient le pouvoir de fixer ces conditions.

FONCTIONNEMENT ET FONCTION DU ROBOTS.TXT

Créé par le développeur Martijn Koster en 1994, dans le cadre de l'instance de l'IETF, le « robots.txt », ou « protocole d'exclusion », permet à un éditeur d'employer une ressource écrite en *American Standard Code for Information Interchange* (ASCII) pour donner des instructions aux agents logiciels appelés *crawlers* ou *spiders*, chargés par leurs concepteurs de consulter et d'enregistrer (c'est-à-dire de « scanner ») le contenu des pages. Positionné à la racine du site web, le robots.txt est par convention le premier élément visité par les *crawlers*. L'éditeur y indique quelles zones de son site doivent être exclues de leur périmètre d'action. Il fixe les limites à l'intérieur desquelles les *crawlers* pourront agir et au-delà desquelles ils ne se rendront pas. Cette *territorialisation* va dans le sens du droit d'auteur, et en particulier de sa composante appelée « droit de divulgation », « *qui donne à l'auteur le pouvoir de décider le moment et le mode de communication des œuvres au public* » (Benhamou et Farchy, 2009, p. 59). Les *crawlers* étant essentiellement utilisés par les moteurs de recherche, le robots.txt est présenté comme un protocole visant à permettre à un éditeur d'exclure certaines pages de l'index des moteurs[4].

Le robots.txt est un « protocole volontaire » (Elmer, 2009, p. 222), fruit d'un accord qui peut à tout moment être violé sans que cela soit nécessairement hors la loi. Il est présenté sur son site officiel comme le fruit d'un « consensus » entre les auteurs du protocole et les personnes concernées par son action[5]. Ce même site précise que le protocole n'appartient pas à une organisation commerciale, que son respect n'est en rien garanti et qu'il faut le considérer comme une « facilité commune » créée pour permettre aux éditeurs de contenus de se protéger contre la venue d'agents non désirés[6].

4. Il existe d'autres types de *crawlers* que ceux des moteurs de recherche, comme *IssueCrawler* et *NaviCrawler*, utilisés par les chercheurs en Humanités Digitales pour récolter des informations à propos des réseaux hypertextes (cf. Rogers, 2013).
5. http://www.robotstxt.org/orig.html.
6. *Ibid*.

Deux commandes principales sont inscrites à l'intérieur du « robots.txt ». La première (« *User-Agent* ») sert à nommer les agents concernés par la ressource. Si la commande est laissée vide, ou remplie par une étoile (« * »), le robots.txt est adressé à l'ensemble des *crawlers* (ce qui ne signifie pas que l'ensemble des agents obéiront à ses directives). La seconde (« *Disallow* ») permet de donner des instructions aux *crawlers* mentionnés par la première. Si elle est laissée vide, le *crawler* comprend qu'il peut scanner l'ensemble du contenu. Ainsi, par défaut, si l'éditeur ne remplit pas le robots.txt, rien ne protégera ses contenus de la venue d'agents logiciels.

> Exemple n° 1 : l'éditeur autorise tous les *crawlers* à scanner toutes les pages
> User-agent: *
> Disallow:
>
> Exemple n° 2 : l'éditeur interdit tous les *crawlers* de scanner toutes les pages
> User-agent: *
> Disallow: /
>
> Exemple n° 3 : l'éditeur exclut les pages du répertoire « /monde » du champ d'indexation de Google
> User-agent: Googlebot
> Disallow: /monde

S'il peut être considéré comme une mise en garde, le robots.txt peut également être perçu comme un appel du pied, dès lors qu'il indique l'emplacement d'informations potentiellement sensibles (Spencer, 2009). Puisque rien n'empêche ou n'interdit d'outrepasser le protocole, n'importe qui pourra en effet décider de récolter l'ensemble des informations mentionnées par le robots.txt. Ainsi, la commande « *Disallow* » agit comme une demande de « bien vouloir » ignorer certains contenus, et peut éventuellement agir comme une indication quant à l'emplacement de contenus sensibles, davantage qu'elle n'institue une quelconque barrière technique.

Le robots.txt, qui ne barre pas techniquement le passage des *crawlers*, a-t-il une fonction juridique ? Intéressé par cette question, Nicklas Lundblad explique que le débat est loin d'être tranché et propose de comparer la valeur légale du robots.txt à celle des conditions d'utilisation des sites web (Lundblad, 2007). Dans de nombreux cas, rappelle-t-il, les cours de justice américaines et européennes ont sanctionné le non-respect des conditions générales d'utilisation (CGU). Ces décisions ont renforcé la doctrine selon laquelle un individu pouvait se rendre coupable d'une violation de propriété

en visitant un site web. Tout en exposant l'argument *ex analogia* selon lequel le robots.txt pourrait éventuellement être considéré comme des CGU spécifiques aux agents de *crawling* automatisé, Lundblad indique que sa comparaison est rendue hasardeuse par l'existence d'une différence significative entre les CGU et le robots.txt, liée au fait que les CGU s'adressent aux individus alors que le robots.txt s'adresse à des agents logiciels. Il est difficile de savoir, dans ces conditions, si oui ou non un *crawler* peut être coupable de violation de propriété au même titre qu'un individu. « *Est-ce qu'un train par exemple peut violer la propriété privée ? Une voiture ? Qu'en est-il, du coup, d'un agent logiciel ?* », s'interroge Lundblad. Quoi qu'il en soit, nous apprenons dans cet article que les conditions requises pour que le protocole d'exclusion agisse ne dépendent pas du seul usage de la syntaxe du ASCII par le codeur, ni des seules actions de celui qui construit une machine destinée à comprendre cette syntaxe et à obéir à ses commandes, mais également des interprétations que la justice est susceptible de donner aux actions qui auront été menées par le codeur et par la machine *grâce* aux lignes de code, d'une part, et, d'autre part, *malgré* les lignes des codes.

Puisque Google publie des documents officiels indiquant que son moteur respecte le robots.txt, Niklas Lundblad explique que cela peut éventuellement doter le protocole d'une valeur juridique. Autrement dit, le code aura une valeur contractuelle si le concepteur d'une machine avec laquelle le protocole est censé permettre de communiquer annonce publiquement que ses machines exécuteront les commandes du protocole. La recommandation faite par Google sous forme de « Conseil aux webmasters » s'ajoute par conséquent aux lignes de code ASCII rédigées par les éditeurs et vient modifier la valeur juridique de la syntaxe en la chargeant de garantir le droit d'auteur.

Tant que le propriétaire d'une machine chargée d'exécuter le code ne se prononce pas officiellement, le rôle du robots.txt n'est donc ni celui d'un contrat qui définirait théoriquement les frontières d'un espace protégé ni celui d'une barrière qui les bloquerait techniquement. Le robots.txt n'a pas une *fonction* de clôture garante de la propriété dès lors que n'importe quel agent logiciel peut *techniquement* l'ignorer, et il n'a pas non plus une *valeur* de clôture, puisque le propriétaire du logiciel qui ne le respecte pas n'est *juridiquement* coupable de rien. L'éditeur qui implémente la ressource sait à quoi s'en tenir : ceux qui voudront désobéir au message rédigé en ASCII dans la syntaxe du protocole d'exclusion y désobéiront sans forcer de barrière ni outrepasser de loi. En revanche, le concepteur des agents, contrairement à l'éditeur, a le

pouvoir de *traduire* la commande « *Disallow* ». Il la transformera en barrière technique s'il paramètre ses agents pour qu'ils *ne puissent pas* franchir la commande, et en un accord juridiquement valable – en un contrat – s'il publie un document d'après lequel ses agents *ne doivent pas* franchir le « *Disallow* ». Le pouvoir de transformer la commande « *Disallow* » en barrière technique et de lui donner une valeur juridique se trouve donc entre les mains de celui qui agit depuis l'extérieur du site web et peut choisir de la nature de sa relation avec l'éditeur. Nous dirons qu'il a le pouvoir de faire en sorte qu'un usage donné de la syntaxe ASCII joue le rôle de « clôture numérique » (c'est-à-dire que nous appelons « clôture numérique » le dispositif permettant *à la fois* de barrer techniquement et de réguler juridiquement l'accès au code par les agents logiciels).

LA CONTROVERSE DE L'*AUTOMATED CONTENT ACCESS PROTOCOL*

Des éditeurs mécontents des conditions et des modalités du protocole d'exclusion ont proposé en 2007 un nouveau protocole exprimé lui aussi en ASCII, mais dans une autre syntaxe. Leur but était de reprendre la main sur le pouvoir de fixation des clôtures numériques. Cependant, les intermédiaires techniques, Google en tête, ont refusé de respecter ce nouveau protocole. Celui-ci, même s'il pouvait (et peut toujours) être utilisé, ne pouvait plus avoir l'effet escompté. Nous reviendrons ci-après sur cet échec qui nous permettra de sonder les coulisses de l'élaboration des protocoles.

L'initiative *Automated Content Access Protocol* (ACAP) fut lancée en novembre 2007 par le *European Publishers Council* (EPC), la *World Association of Newspapers* (WAN) et l'*International Publishers Association* (IPA) dans le but que les éditeurs puissent « communiquer leurs politiques d'accès et d'utilisation de leurs contenus dans une forme *machine-readable* aux moteurs de recherche et aux agrégateurs » (ACAP, 2009, p. 1). 1 250 éditeurs signataires se sont joints à l'initiative, rassemblés en un même réseau d'influence, afin d'être entendus par des intermédiaires techniques d'envergure mondiale. En France, le Syndicat de la Presse Quotidienne Nationale et le Syndicat de la Presse Quotidienne Régionale sont membres de la WAN, et donc signataires de l'ACAP, tandis que le Syndicat National de l'Édition est membre de l'IPA. L'Agence France Presse fut quant à elle l'un des pilotes du projet. Les éditeurs ne sont pas passés par les instances de standardisation des protocoles. On aurait pu imaginer qu'ils proposent leur projet à l'IETF

ou au W3C, mais ils ne l'ont pas fait. Même s'ils ont invité certains acteurs, et notamment Google, à donner leur avis concernant leur projet, les éditeurs n'ont pas souhaité ouvrir les négociations, attachés à un programme d'action selon lequel eux seuls devraient décider de ce qui pouvait être fait de leurs contenus, et comment, plutôt qu'au programme d'action « classique » des acteurs de l'Internet, selon lequel tous les acteurs concernés peuvent avoir leur mot à dire concernant un nouveau protocole.

ACAP est un protocole non propriétaire destiné à adapter le web aux lois en vigueur en termes de *copyright* et de droit d'auteur. Sur le site dédié à l'initiative, le problème auquel ACAP prétend répondre est présenté comme découlant du fait que les conditions générales d'utilisation d'un site (CGU) se trouvent habituellement sur une page que les agents de *crawling* automatisé ne peuvent pas lire et, donc, que les machines ne pourront pas traduire. ACAP propose par conséquent aux éditeurs une syntaxe dont la raison d'être est de leur permettre de traduire leurs CGU dans un langage et en un lieu qui leur permettront à la fois de contrôler la diffusion de leurs contenus et de « maximiser les bénéfices de [leur] relation avec les moteurs et les agrégateurs » (ACAP, FAQ). Une telle possibilité résoudrait le problème exposé ci-avant à partir des travaux de Niklas Lundblad (2007) : les CGU seraient exprimées *à la fois* dans une langue adressée aux individus et dans un langage adressé aux machines, en deux lieux distincts (une page web pour les CGU, la racine du site pour le protocole), et ne concerneraient dès lors plus les seuls individus, mais les utilisateurs au sens large, de manière à rendre indissociables du point de vue juridique les rapports noués par l'éditeur avec des individus et avec des agents logiciels.

Les éditeurs signataires de l'ACAP estiment que le protocole d'exclusion relève d'une interprétation trop libérale du droit d'auteur, et que les éditeurs n'ont pas les moyens de faire en sorte que leurs décisions soient respectées. Dès lors, il leur a semblé nécessaire de remédier à la situation grâce à un nouveau protocole sans passer par l'IETF ou le W3C. Dans la citation ci-dessous, nous voyons comment une montée en généralité a été opérée, visant à représenter les intérêts de l'ensemble des éditeurs de contenus :

« *La loi du copyright existe dans le monde entier et donne aux créateurs et aux éditeurs le droit de décider comment le contenu qu'ils ont créé et pour lequel ils ont investi devrait être légalement exploité par les autres. Les industries médiatiques, qui existent seulement grâce au copyright, contribuent massivement à développer l'économie du monde physique et sont vitales pour*

le futur économique. La possibilité d'exprimer et de partager des permissions pour l'accès et l'usage [des contenus] de manière standardisée est une part de l'infrastructure nécessaire [à l'activité des industries médiatiques] tant sur le réseau que dans le monde physique. » (ACAP, FAQ)

Les défenseurs de l'ACAP tentent par ailleurs de désamorcer l'argument de ceux qui pourraient s'opposer à leur projet en leur reprochant de vouloir nuire à l'accessibilité des contenus. À ces derniers, Dominic Young, directeur des services d'édition de *News International*, explique :

« *Le copyright a conduit non pas à une restriction des contenus, mais à une explosion, un constant et effréné partage des idées ainsi qu'un raz-de-marée de choix pour les consommateurs. Si l'économie du savoir est la clef de voûte de la croissance future, alors le copyright en est la fondation.* » (ACAP, FAQ)

Cette citation fournit des éléments susceptibles d'expliquer pourquoi les éditeurs signataires du ACAP ne se sont pas tournés vers le W3C ou l'IETF pour proposer leur protocole. Ils auraient risqué d'y rencontrer une forte opposition de la part d'acteurs attachés à la facilité de circulation des contenus plutôt qu'à la protection des ayants droit. Ce fut par exemple le cas en 2013 lorsque fut proposé d'ajouter l'*Encrypted Media Extensions* (EME) aux standards du HTML 5, alors que la négociation était en cours au W3C. L'EME vise à permettre à un éditeur de mettre en place un système de *Digital Right Management* (DRM) en exigeant la présence d'un logiciel tierce ou d'un matériel spécifique pour lire le contenu. L'*Electronic Frontier Foundation* s'est opposé vent debout à l'EME, accusant ses défenseurs (parmi lesquels figuraient pourtant Google) d'agir pour « a*paiser Hollywood qui est mécontenté par Internet depuis presque aussi longtemps qu'existe le Web et qui a toujours demandé à ce qu'on lui fournisse des infrastructures techniques évoluées pour contrôler le fonctionnement des ordinateurs de son public* »[7]. Selon l'EFF, il s'agissait d'une « *terrible erreur pour la communauté du Web de laisser la porte ouverte à l'infection des standards du W3C par la gangrène culturelle anti-technologique d'Hollywood* »[8]. Finalement, l'EME est passé au statut d'ébauche public et suit la voie de la standardisation au W3C. Ce qui nous intéresse ici, c'est d'une part de constater que les signataires de l'ACAP se seraient sans nul doute heurtés aux mêmes arguments que ceux opposés par l'EFF à l'EME s'ils

7. http://www.01net.com/editorial/589389/au-sein-du-w3c-google-et-microsoft-travaillent-a-faire-des-drm-un-standard-du-web/
8. *Ibid.*

avaient tenté de passer par le W3C ou l'IETF. D'autre part, l'EME est soutenu, entre autres, par Microsoft, Google et Netflix, qui ont un poids certain dans les négociations menées au W3C. Sans de tels appuis, le protocole ACAP aurait probablement échoué au W3C, ou aurait en tout cas dû être revu, corrigé et amendé étant donné les contre-arguments qui auraient été avancés.

En outre, le robots.txt existait depuis plus de dix ans lorsque l'initiative ACAP fut lancée, ce qui risquait de compliquer le processus d'adhésion massive au nouveau protocole. En effet, comme le souligne Laura DeNardis, « *une fois que [les protocoles] sont largement implémentés, des forces sociales et économiques considérables sont nécessaires pour supplanter les standards établis* » (DeNardis, 2014, p. 65). La plupart des *crawlers* avaient été paramétrés pour prendre en compte le robots.txt, tandis que de très nombreux webmestres s'étaient habitués à l'utilisation d'un protocole auquel ils ne reprochaient rien. C'est pourquoi l'ACAP fut créé de manière à ce que la nouvelle syntaxe se greffe à la syntaxe existante, implémentée à l'intérieur du robots.txt, à la racine du site, ou bien dans les balises métatags, sur les pages individuelles, et rédigée elle aussi en ASCII. Il s'agissait d'une spécification de la syntaxe visant à exprimer « des permissions plus précises et plus nuancées » (ACAP, 2009, p. 2). Sur le site du ACAP, cette greffe est justifiée étant donné « l'insistance des moteurs de recherche ». Le protocole est présenté comme la « version 2.0 » du robots.txt, ses signataires souhaitant compléter et préciser l'existant plutôt que de le concurrencer.

En résumé, le projet ACAP constitue une tentative de la part des éditeurs pour s'approprier le pouvoir de fixation des clôtures numériques du point de vue juridique d'une part, en généralisant la notion d'« utilisateurs » et en traduisant leurs CGU en ASCII de manière à faire respecter le droit d'auteur par les machines, et du point de vue technique d'autre part en rendant compatible le ACAP avec les dispositions du robots.txt tout en demandant aux intermédiaires de paramétrer leurs agents pour qu'ils ne puissent pas ignorer les commandes du ACAP.

ANALYSE COMPARATIVE DES PRINCIPALES FONCTIONS DU ROBOTS.TXT ET DE L'ACAP

Les documents de l'ACAP stipulent que le nouveau protocole permet de faire tout ce que le robots.txt permet lui aussi de faire, mais que ce sera signifié désormais dans les « termes ACAP ». Nous retrouvons ici un élément qu'il

est intéressant de rapprocher de ce que dit Mark Marino lorsqu'il présente le projet des *critical code studies*, à savoir qu'il s'agit moins de porter attention à ce que le code fait qu'aux choix effectués par le codeur parmi différentes possibilités qui auraient toutes conduit aux mêmes actions (Marino, 2006). En effet, le code lui-même, en tant que texte, *dit* quelque chose, et ce n'est pas parce que l'ACAP peut techniquement réaliser des actions semblables à celles du protocole d'exclusion que les deux syntaxes ont la même *signification*, notamment d'un point de vue juridique. Par ailleurs, le guide d'utilisation indique que l'implémentation du ACAP, en plus de préciser et de nuancer les conditions de reproduction et de diffusion d'un contenu propriétaire, a pour objectif de diminuer le nombre d'opérations nécessaires à l'édition du fichier robots.txt. Ce serait donc également une mesure de simplification technique du point de vue des éditeurs (nous verrons toutefois qu'il ne s'agit pas d'une simplification technique du point de vue des moteurs). Pour résumer, les défenseurs de la syntaxe ACAP prétendent augmenter la force juridique de la syntaxe ASCII, tout en simplifiant ses conditions d'implémentation technique. Ci-après, nous analysons les commandes principales de la syntaxe ACAP en les comparant aux commandes de la syntaxe usuelle du robots.txt, de manière à identifier certaines différences *significatives* entre les deux protocoles.

Allow/Disallow

ACAP permet de mentionner à quel(s) agent(s) logiciel(s) une commande en particulier est destinée. Il est possible de s'adresser à tous les *crawlers*, à un seul d'entre eux ou bien à un groupe. C'était déjà possible avec le robots.txt grâce à la fonction « *User agent* » mais, dans le cas d'ACAP, cela permet de prévenir le *crawler* dès son arrivée sur le site que le fichier robots.txt, contient deux syntaxes : l'une classique, destinée aux intermédiaires qui ne comprennent pas les commandes du ACAP, et l'autre destinée à ceux qui lisent et respectent les consignes du ACAP. Il est ainsi possible d'interdire aux *crawlers* qui ne respecteraient pas le ACAP de scanner les contenus tandis que les *crawlers* qui le comprennent et le respectent y auront, eux, accès. De la sorte, les éditeurs peuvent faire pression sur les propriétaires des *crawlers* en avantageant ceux qui adhèrent à leurs projets. Voici ce qu'on trouvera dans le robots.txt :

```
User-agent: *                              #Syntaxe usuelle
Disallow: /
ACAP-ignore-conventional-records           #Syntaxe ACAP
ACAP-crawler: *
```

ACAP-allow-crawl: /

La différence majeure entre la syntaxe de l'ACAP et celle de la syntaxe usuelle du robots.txt réside dans le fait que cette dernière permet d'interdire l'accès à l'ensemble du site, à un répertoire ou à un document, tandis que la syntaxe de l'ACAP permet d'autoriser l'accès à un site, un répertoire ou un document. Le robots.txt se base essentiellement sur la commande « *Disallow* »[9], alors que l'ACAP se base sur la commande « *Allow* ». Ainsi, tandis que le robots.txt est prohibitif et tacite (dès lors que je ne dis rien, c'est que je suis d'accord), le second est permissif et explicite (je dis que je suis d'accord). L'ACAP n'est plus seulement un protocole d'exclusion mais aussi, et avant tout, un protocole d'inclusion. Plutôt que de dire aux *crawlers* les actions qu'ils ne peuvent pas faire, il leur dit ce qu'ils auront le droit de faire. L'absence de commande serait alors l'équivalent d'une clôture numérique, alors que dans le cas du protocole d'exclusion seule la commande « *Disallow* » peut se voir déléguer ce rôle. Autrement dit, ce qui change avec le ACAP, c'est que l'éditeur n'a plus besoin d'agir pour faire respecter son droit d'auteur : la clôture est construite autour de son site par défaut. L'éditeur peut ensuite décider de l'abattre en autorisant explicitement le franchissement de telle ou telle clôture par tel ou tel *crawler*. C'est plus contraignant pour le *crawler*, qui se trouve dans une situation où il ne peut ni ne doit franchir les clôtures que s'il en a explicitement reçu l'autorisation.

Cependant, un élément essentiel ne change pas dans le cas de la syntaxe ACAP : comme avec la syntaxe usuelle du robots.txt, seul l'intermédiaire peut déléguer au protocole une valeur juridique et technique. Autrement dit, seuls ceux qui paramètrent les *crawlers* ont le pouvoir d'*institutionnaliser* la syntaxe ASCII prévue par l'ACAP. S'ils ne changent pas le fonctionnement de leurs *crawlers* et s'ils ne communiquent pas officiellement leur intention de respecter le protocole ACAP, ce dernier ne sera plus qu'une « mise en garde » que les intermédiaires techniques ignoreront sans craindre de sanction.

Usage purpose

Si nous nous analysons les autres fonctions du ACAP, nous apprenons qu'il est possible pour un éditeur, dans le cas où un *crawler* œuvrerait pour le compte de plusieurs moteurs, de choisir sur quel moteur il souhaite apparaître

9. Une commande « Allow » existe pour le robots.txt, mais elle est très peu utilisée (cf. Spencer, 2009).

et sur quel moteur il ne veut pas être référencé. Par exemple, si un éditeur veut être indexé sur le moteur généraliste, mais pas sur les moteurs verticaux « images » et « news », il spécifie :

```
ACAP-crawler: *
ACAP-allow-crawl: /
ACAP-crawler: SearchBot1
ACAP-usage-purpose: news.*
ACAP-usage-purpose: images.*
ACAP-disallow-crawl:
```

La Fédération italienne des éditeurs de journaux (Fieg) a porté plainte contre Google en août 2009 auprès de l'Autorité italienne de la concurrence pour abus de position dominante (Macchiati, 2010). L'un des reproches adressés par la presse à Google était de contraindre les éditeurs désireux d'être indexés sur le moteur généraliste Google Search à accepter d'intégrer le champ d'action du moteur spécialisé Google Actualités. Selon la Fieg, lorsqu'un éditeur de presse reconnu indiquait à Google qu'il ne souhaitait pas intégrer le champ du moteur vertical, la firme lui signifiait que, dans ce cas, il ne pourrait pas intégrer non plus le champ du moteur généraliste. Or il se trouve que la fonction « *Usage purpose* » du ACAP permettrait précisément d'indiquer à Google qu'on souhaite figurer dans l'index de son moteur généraliste mais qu'on veut être exclu de l'index de ses moteurs spécialisés. C'est la traduction, dans la syntaxe ASCII, d'un ordre que les éditeurs souhaitent intimer à Google : « indexe-moi dans ce moteur, ne m'indexe pas dans celui-là ».

Follow

La commande « *follow* » de la syntaxe ACAP permet de dire si des liens doivent être suivis ou non. Il est possible normalement de procéder à une telle action grâce à des métatags HTML spécifiques implémentées sur les pages, mais il n'est pas possible de le faire directement depuis la racine du site, grâce à la syntaxe ASCII du robots.txt. Une des particularités d'ACAP est donc de permettre à l'éditeur de donner cette information au *crawler* dès la racine du site en ajoutant une commande à la ressource robots.txt. Dans l'exemple ci-après, les *crawlers* ont l'autorisation de visiter le seul contenu du répertoire « /public » (le signe « $ » annonce une exception) mais ne peuvent pas, dans ce répertoire, suivre les liens du sous-répertoire « /comments ».

```
ACAP-crawler: *
ACAP-disallow-crawl: /
ACAP-allow-crawl: /$
ACAP-allow-crawl: /public/
ACAP-disallow-follow: /public/comments/
```

Dans le cas du robots.txt, qui ne prévoit pas de commande « *allow* », cela aurait dû être fait en deux étapes : d'abord communiquer au *crawler* qu'il ne pouvait pas scanner les autres répertoires, mentionnés alors un à un, excepté « /public », puis implémenter une balise méta-tag HTML à la racine du sous-répertoire « /public/comments » (<meta name=»robots» content=»noindex, nofollow»>).

Index

La commande « *index* » du ACAP permet de spécifier la zone d'indexation, non pas seulement comme le robots.txt et les méta-tags usuels en la diminuant, mais également en la désignant explicitement. Par exemple, dans le cas ci-dessous, où nous reprenons et enrichissons l'exemple donné ci-avant, seules les pages du répertoire « /public » peuvent être *crawlées* et indexées, tandis qu'aucune autre action que le *crawl* et l'indexation n'est autorisée : à la dernière ligne, « *disallow-other* » signifie que toutes les actions qui ne sont pas explicitement permises sont interdites.

```
ACAP-crawler: *
ACAP-disallow-crawl: /
ACAP-allow-crawl: /$
ACAP-allow-crawl: /public/
ACAP-allow-index: /public/
ACAP-disallow-other: /public/
```

Il est également possible de spécifier les formats qu'on ne veut pas indexer. Dans ce cas, par exemple pour les images aux formats « .jpg » et « .gif », il suffira d'ajouter :

```
ACAP-disallow-index: /public/*.jpg$
ACAP-disallow-index: /public/*.gif$
```

Enfin, il est possible de spécifier explicitement le format que le *crawler* est autorisé à indexer. Cela devient une permission et un accord explicite plutôt

qu'une absence d'interdiction et un accord tacite. Il suffit pour cela de permettre le *crawling* et d'interdire l'indexation des documents tout en permettant explicitement la seule indexation des documents au format, par exemple, HTML, ce qui donne :

```
ACAP-allow-crawl: /public/
ACAP-disallow-index: /public/
ACAP-allow-index: /public/*.html
```

Les autres commandes de la syntaxe ACAP fonctionnent toutes sur le modèle des commandes décrites ci-avant. Il s'agit, à chaque fois, d'affiner la granularité du champ de manœuvres de l'éditeur et de rendre l'accord explicite. Il existe des commandes permettant de limiter la date de disponibilité du contenu, de spécifier les attributs des éléments à *crawler*, de limiter le nombre de mots du « *snippet* », de prohiber le changement de format parfois effectué par les moteurs (souvent un PDF transformé en HTML), et d'interdire toute modification typographique, toute traduction, toute annotation et toute ré-éditorialisation[10]. Une telle granularité permet aux éditeurs de contrôler extrêmement finement les modalités du processus d'intermédiation. Leurs conditions sont ainsi rédigées dans un langage approprié, et le *crawling* soumis à des clauses explicites.

En regardant de près les deux protocoles, on s'aperçoit que certaines commandes de l'ACAP n'ont pas d'équivalent dans la syntaxe du robots.txt. Notamment, le ACAP prévoit une commande « *cloaking* », censée permettre à un éditeur de montrer autre chose au *crawler* que ce que voit l'internaute quand il se rend sur la page. Cela permettrait à un éditeur d'un site pornographique de « faire croire » au moteur de recherche que son site est consacré à la botanique. Parce qu'une telle technique risquerait d'aboutir à la multiplication des liens non pertinents dans les listes de résultats, les moteurs sont farouchement opposés au « *cloaking* » (Sullivan, 2009b).

Finalement, presque tous les concepteurs et propriétaires de moteurs de recherche ont refusé de paramétrer leurs machines, de manière à prendre en compte la syntaxe ACAP. Ainsi, le projet d'une entente explicite entre les moteurs et les éditeurs n'a jamais vu le jour, car les conditions n'ont pas été

10. Dans le cas d'une interdiction de ré-éditorialisation, si une version « en cache » est autorisée, le moteur devra publier le texte brut, comme il a été mis en ligne sur ses serveurs, sans aucune remise en forme.

réunies qui auraient permis à la syntaxe ACAP de provoquer l'adhésion de ceux à qui elle était censée être adressée.

L'INCLUSION EXCLUE, OU LE POUVOIR DE CELUI QUI TRADUIT

Le moteur de recherche Exalead, dont les parts de marché sont très faibles, est le seul à avoir participé au projet ACAP en qualité de pilote. La firme Google, ayant pourtant été consultée lors de l'élaboration du protocole, a finalement rejeté la proposition en prétextant qu'elle était techniquement trop difficile à mettre en place, sans préciser pour quelle raison spécifique. Les éditeurs signataires, quant à eux, soutiennent qu'elle est techniquement réaliste. Le déséquilibre dans les rapports de forces entre les intermédiaires techniques, Google en particulier, et les éditeurs fut à cette occasion clairement visible :

« *Les moteurs de recherche sont une sorte de* gatekeepers, *parce que ce qu'ils décident de promouvoir devient effectivement une "loi". S'ils ne promeuvent pas une commande d'exclusion en particulier, cela revient à ce que celle-ci n'existe pas*. » (Sullivan, 2007)

La plupart des observateurs et des experts du domaine se sont montrés très sceptiques vis-à-vis du ACAP, notamment car, selon eux, le protocole était aussi facile à mettre en place pour l'éditeur du contenu (le codeur) qu'il était compliqué à mettre en œuvre pour les propriétaires des agents logiciels (les concepteurs de machines à traduire le code). Ces derniers auraient en effet dû reparamétrer leurs *crawlers* dans le but de s'adapter à la syntaxe ACAP, ce qui les aurait placés selon leurs dires en face de difficultés indépassables. Ainsi, les éditeurs, soucieux de leur propre programme d'action, et désireux de proposer un protocole sans passer par les instances de standardisation, n'auraient pas assez considéré les contraintes des concepteurs des agents. La nouvelle syntaxe de l'ASCII a par conséquent bel et bien été créée et mise en service, mais n'a pas reçu l'adhésion des acteurs sans l'écoute desquels, pourtant, ce qu'elle *dit* ne *fait* plus rien. Comme le dit le guide d'implémentation lui-même :

« *Pour qu'ACAP fonctionne comme prévu, les opérateurs doivent reprogrammer volontairement leurs agents afin d'interpréter les expressions ACAP présentes dans le robots.txt et intégrées au contenu web. La façon dont un agent interprète ACAP dépendra de la manière dont cette reprogrammation aura eu lieu* » (ACAP, 2009, p. 4).

D'après l'avis de plusieurs éditeurs l'ayant expérimenté, la syntaxe de l'ACAP ne fonctionnait pas ou très mal. Il ne faut donc pas considérer simplement que les intermédiaires techniques n'ont pas voulu de l'ACAP, mais également que le protocole présentait des difficultés majeures en termes d'implémentation. Notamment, il semblerait que sous prétexte de donner aux éditeurs la plus forte granularité possible en termes de possibilités d'action, le protocole ait abouti à un « étalage vertigineux » de commandes complexes à mettre en œuvre (Sullivan, 2009b). Rob Jonas, employé par Google en qualité de directeur des partenariats « media & édition » en zone Europe, s'est prononcé en mars 2008 afin d'expliquer pourquoi l'ACAP ne lui paraissait pas convenir, et pourquoi le robots.txt suffisait amplement :

« *Le point de vue général est que le protocole robots.txt permet à la plupart des éditeurs de faire ce qu'ils doivent faire. Tant que nous ne considérons pas qu'il existe des raisons convaincantes pour l'améliorer, nous pensons que ce protocole permettra à chacun d'agir comme il l'entend.* » (Oliver, 2008a)

Dans cette citation, nous voyons, d'une part, la montée en généralité que tente d'effectuer le porte-parole : en évoquant « le point de vue général » et non de celui de Google en particulier, il sous-entend que l'opinion de Google est conforme à celui de la majorité. Les signataires de l'ACAP se heurtent à une critique qui ne leur aurait sans doute pas été opposée en ces termes s'ils étaient passés par le W3C. D'autre part, le porte-parole de Google prétend que le robots.txt fournit les outils dont les éditeurs « ont besoin ». Or le projet d'ACAP n'est pas tant de remplacer les commandes du robots.txt, dont l'utilisation peut en effet à aboutir au même résultat, que de rendre explicites les termes de la relation nouée entre l'éditeur et le propriétaire d'un agent de *crawling*. Nous voyons également dans cette citation que le pouvoir d'instituer des lignes de code en tant que clôture effective appartient à l'intermédiaire qui, tant qu'il ne verra pas de raisons pour changer de protocole (c'est-à-dire tant qu'il n'aura pas intérêt à le faire), fera en sorte de maintenir une syntaxe en ligne avec ses propres intérêts.

Gavin O'Reilly, président de WAN et fervent défenseur du ACAP, a répondu à Jonas :

« *C'est plutôt étrange de la part de Google de dire aux éditeurs ce qu'ils devraient penser à propos du robots.txt, lorsque les éditeurs du monde entier – de tous les secteurs – ont déjà et clairement dit à Google qu'ils n'étaient absolument pas d'accord.* […] *Si la raison pour laquelle Google ne supporte pas (apparemment) ACAP est basée sur ses propres intérêts commerciaux,*

alors la firme devrait le dire, plutôt que de remettre en cause la confiance qu'on peut nous faire. […] *Google devrait réfléchir au fait qu'après douze mois de réflexions intensives et internationales et un développement actif – auquel Google a également pris part – les éditeurs n'ont pas seulement souligné les inadéquations du robots.txt, mais ont construit une solution pratique, ouverte et opérationnelle destinée aux éditeurs et aux agrégateurs.* […] *Par conséquent – une fois de plus –, nous appelons Google à adhérer à ACAP et à reconnaître le droit qu'ont les propriétaires de contenus de déterminer comment leurs contenus seront utilisés.* » (Oliver, 2008b)

Gavin O'Reilly se pose en porte-parole de l'ensemble des éditeurs de contenus concernés par l'action de Google. Il remet en cause la montée en généralité opérée par Google, arguant que ses porte-parole ne peuvent pas déterminer ce que les éditeurs doivent penser. Il pointe du doigt les opportunités commerciales en sous-entendant que Google défend ses intérêts et non ceux des éditeurs. Enfin, Gavin O'Reilly en vient aux arguments techniques, prétendant que la solution proposée fonctionne et que c'est à Google de déléguer aux éditeurs le pouvoir de fixation du protocole, et non l'inverse. En d'autres termes, il revendique le pouvoir de décider de quelle façon et à quelles conditions une certaine utilisation de l'ASCII peut avoir la valeur juridique et technique d'une clôture garante du droit de propriété.

Eric Schmidt, président de Google, a souhaité apaiser la controverse en précisant pour sa part qu'il s'agissait uniquement d'un problème technique, et non, pour Google, d'une volonté de voler « le contrôle de l'information » aux éditeurs :

« *ACAP est un standard proposé par un groupe de personnes qui essaient de résoudre un problème. Certains de nos employés travaillent avec eux pour voir si la proposition peut être modifiée de manière à pouvoir fonctionner sans remettre en cause le fonctionnement du moteur de recherche. Or, pour l'instant, [ACAP] est incompatible avec la façon dont notre système opère.* » (Corner, 2008)

Aujourd'hui, ACAP est au point mort. En 2011, à Berlin, le projet a été repris par l'*International Press and Telecommunications Council* (IPTC), consortium d'agences de presse et d'éditeurs de presse en ligne. Son objectif est désormais d'influencer les projets de loi et les décisions prises par la Commission européenne en matière de protection du droit d'auteur. D'une

proposition technique qui se voulait novatrice mais a été taxée d'irréalisme, le projet a glissé vers une stratégie de lobbying politique. Ainsi, même si l'ACAP n'est plus à l'ordre du jour, il a été intégré à un consortium qui, lui, est encore actif. Une version 2.0 d'ACAP a tout de même été définie en juin 2011 à partir du standard *Open Digital Rights Language*, à laquelle, sans surprise, les intermédiaires techniques n'ont pas adhéré non plus.

En janvier 2012, nous avons nous-même questionné à propos du ACAP celui qui chez Google a remplacé Rob Jonas au poste de directeur des partenariats « media & édition » en zone Europe : Madhav Chinnappa. Il nous a répondu que d'après lui le protocole d'exclusion robots.txt fonctionnait très bien et qu'il n'y avait pas de raison de le remplacer. Hélas, il n'a pas souhaité nous en dire davantage de ce sujet. La politique de communication externe de Google étant extrêmement verrouillée, nous n'avons pas réussi à en apprendre plus. Nous avons simplement pu constater que les propos de Madhav Chinnappa étaient identiques à ceux de Rob Jonas.

Il faut noter enfin que s'ils étaient passés par une instance de standardisation, cela aurait peut-être abouti de la même façon. En effet, les standards proposés par l'IETF et le W3C peuvent très bien se faire concurrence (DeNardis, 2014, p. 72). Dans le cas où ACAP aurait été validé comme tel par le W3C, les intermédiaires techniques auraient tout aussi bien pu choisir de ne pas le respecter, et de rester fidèles au protocole d'exclusion. Que le protocole passe ou non par une instance de standardisation, le pouvoir de fixer la syntaxe qui prévaut à l'élévation de barrières techniques et juridiques limitant l'action des moteurs de recherche appartient donc clairement à Google, dès lors que la firme a la capacité de « geler un ensemble spécifique de règles et de relations de pouvoir […], favorisant certaines tâches et certains acteurs par rapport aux autres » (Grimmelmann, 2014, p. 10).

CONCLUSION : LANGAGE AUTORISÉ, CONTRAT LÉONIN

En étudiant comment l'utilisation de l'*American Standard Code for Information Interchange* pouvait barrer techniquement et juridiquement la route des agents de *crawling*, nous en sommes venus à la conclusion que le pouvoir de fixer un protocole sur le Web appartenaient davantage aux concepteurs des machines à traduire le code qu'aux éditeurs qui codent. Le robots.txt est en effet capable de formaliser des relations institutionnelles et économiques

(Elmer, 2009, p. 226), à la seule condition que les intermédiaires techniques acceptent les tenants et les aboutissants d'une telle formalisation. Cette acceptation passe par le paramétrage des agents logiciels destinés à exécuter le code et par la publication officielle d'un document où l'intermédiaire s'engage à respecter les commandes prévues par le protocole. C'est à cette seule condition que le recours à la syntaxe pourra avoir l'effet escompté.

Les éditeurs signataires de l'ACAP ont tenté de reprendre la main sur la syntaxe du ASCII afin que des « clôtures numériques », garantes de la propriété de l'auteur, puissent exister par défaut plutôt qu'*a posteriori*. Ils l'ont fait sans passer par le W3C, ni par aucune instance de standardisation, si bien que les négociations qui ont prévalu pour établir le protocole n'ont pas été ouvertes et que les éditeurs ont essayé d'imposer leur propre protocole. L'ACAP consistait à élargir le périmètre de la notion d'utilisateurs de manière à y intégrer les logiciels, et à renforcer le pouvoir de contrôle des éditeurs. Cependant, la nouvelle syntaxe est restée lettre morte, parce que les intermédiaires techniques, Google en tête, ont refusé d'annoncer officiellement qu'ils la respecteraient et parce qu'ils n'ont pas paramétré leurs *crawlers* dans le but qu'ils la respectent. Cela aurait tout aussi bien pu être le cas si les éditeurs étaient passés par le W3C, les recommandations du consortium n'ayant pas non plus d'autre valeur technique ou juridique que celle d'un consensus. Cependant, au W3C, les éditeurs se seraient opposés à des arguments semblables à ceux que l'EFF a opposés au EME en reprochant ce dernier de vouloir garantir le droit d'auteur au détriment de la libre circulation des contenus. Les difficultés d'implémentation technique du ACAP pour les concepteurs de *crawlers* auraient peut-être été atténuées tandis que certaines commandes comme le « *cloaking* » auraient sans doute dû être abandonnées, dès lors que les intermédiaires auraient pu clairement se prononcer à leur sujet.

Sur le Web, les éditeurs sont très nombreux, ne parlent pas la même langue et leurs activités et leurs objectifs sont extrêmement hétérogènes. Il est bien difficile, dans ces conditions, d'être entendu et d'imposer des modalités de fixation de démarcations qui garantiraient la propriété d'un auteur. Du côté des intermédiaires, en revanche, le marché se concentre autour de quelques acteurs à l'envergure mondiale, dont les outils entrent « mécaniquement en conflit avec le droit d'auteur » (Manara, 2011, p. 147). Il n'est donc pas surprenant que les intermédiaires aient le pouvoir de décider de la syntaxe qui aura cours et qu'ils choisissent que celle-ci soit la moins contraignante possible du point de vue juridique et technique. Les mots de James Grimmelmann

trouvent dans notre cas un écho particulier : « *Pas de consultation, pas de vote, pas de mise en garde, pas de pourvoi en appel, pas de remboursement. Le pouvoir technique peut être exercé en dehors des garde-fous et des contre-pouvoirs qui existent habituellement dans toute démocratie digne de ce nom* » (Grimmelmann, 2014, p. 12).

Finalement, nos observations conduisent à penser la relation des intermédiaires techniques et des éditeurs comme un contrat unilatéral au sein duquel « *une ou plusieurs personnes sont obligées envers une ou plusieurs autres, sans que de la part de ces dernières il y ait engagement* »[11]. L'obligation se trouve du seul côté des éditeurs, qui doivent accepter la syntaxe et les conditions d'utilisation proposées par l'intermédiaire, lequel n'est en rien obligé d'accepter la syntaxe et les conditions que les éditeurs lui soumettent. Dès lors que ces derniers n'implémentent pas le protocole d'exclusion, l'intermédiaire estimera qu'ils sont d'accord pour que leurs documents soient référencés selon des modalités implicites et définies par lui. Le consensus, ici, est *léonin*. Cet adjectif, appliqué à un contrat, désigne la différence de poids des parties dans les négociations dont l'entente est le résultat. Il est clair en effet que les dirigeants et les employés de Google sont ceux qui choisissent les conditions et les modalités du référencement sur leurs services. En reprenant les termes de la sociologie de la traduction, nous dirions que les éditeurs *contractants* peinent à jouer un rôle de *contre-actants* vis-à-vis du programme d'action de Google. Finalement, la plupart des éditeurs n'utilisent pas la syntaxe robots.txt et n'implémentent pas non plus ACAP, dont les moteurs ne comprennent pas la syntaxe. Aussi acceptent-ils, bon gré mal gré, par la médiation (ou l'absence de médiation) du code informatique, les dispositions d'un accord unilatéral et léonin.

11. *Code civil*, 1804, art. 1101-1102, p. 200.

RÉFÉRENCES

ACAP (2009), *Guide to implementation of ACAP Version 1.1 Communication with Crawlers. A component of the ACAP Technical Framework*, issue 1, 9 octobre.

AKRICH M. (1987), « Comment décrire les objets techniques ? », *Techniques & Culture*, vol. 1, n° 54-55, pp. 205-219.

BENHAMOU F., FARCHY J. (2007), *Droit d'auteur et copyright*, Paris, La Découverte, 124 p.

BERRY D. M. (2011), *The Philosophy of Software. Code and Mediation in the Digital Age*, Palgrave Macmillan.

CAYLEY J. (2002), « The Code is not the Text (unless it is the text) », *Electronic Book Review*.

CERTEAU M. (de) (1980), *L'invention du quotidien, 1. Arts de faire*, Paris, UGE/10-18, 348 p.

COHEN J. L. (2009), « Working with News Publishers », *Google public policy blog*.

CORNER S. (2008), « ACAP content protection protocol "doesn't work" says Google CEO », *ITWire*.

COUTURE S. (2012), « L'écriture collective du code source informatique. Le cas du commit comme acte d'écriture », *Revue d'anthropologie des connaissances*, vol. 6, n° 1, pp. 21-42.

COX G., McLEAN A., WARD A. (2001), « The Aesthetics of Generative Code », *Generative Art 00 conference. Politecnico di Milano, Italy*.

CRAMER F. (2001), « Digital Code and Literary Text », *Beehive*, 4, 3.

CRAMER F. (2005), *Words Made Flesh. Code, Culture, Imagination*, Rotterdam, Media Design Research Piet Zwart Institute.

DeNARDIS L. (2014), *The Global War for Internet Governance*, New Haven, Yale University Press.

ELMER G. (2009), « Robots.txt: The Politics of Search Engine Exclusion », in J. PARIKKA, T. SAMPSON (dir.), *The Spam Book: On Anomalous Objects of Digital Culture*, New York, Hampton Press, 330 p.

FULLER M. (2003) *Behind the Blip: Essays on the culture of software*, New York, Autonomedia.

FULLER M. (dir.) (2008), *Software studies/a lexicon*, Cambridge, MIT Press, 334 p.

FULLER M., MARINO M. (2013), « Matthew Fuller in Conversation with Mark Marino », *e-media studies*, vol. 3, n° 1.

GLAZIER L. P. (2006), « Code As Language », *Leonardo Electronic Almanac*, vol. 14, n° 5.

GRIMMELMANN J. (2014), « Anarchy, Status Updates, and Utopia », *Pace Law Review*, Forthcoming, University of Maryland Legal Studies Research Paper n° 4.

LATOUR B. (1999), *Pandora's Hope. Essays on the Reality of Science Studies.* Cambridge MA/London, Havard University Press, 336 p.

LESSIG L. (2000), « Code is law. On Liberty in Cyberspace », *Harvard Magazine*, traduction française sur *Framablog.*

LESSIG L. (2006), *Code 2.0*, New York, Basic Books, 410 p.

LEVY S. (2011), *In The Plex: How Google Thinks, Works, and Shapes Our Lives*, New York, Simon & Schuster, 432 p.

LUNDBLAD N. (2007), « e-Exclusion and Bot Rights: Legal aspects of the robots exclusion standard for public agencies and other public sector bodies with Swedish examples », *First Monday*, vol. 12, n° 8.

MACCHIATI A. (2010), « I motori di ricerca su Internet e il mercato delle news. Profili antitrust e regolamentari », *Mercato Concorrenza Regole*, vol. 3, pp. 469-490.

MACKENZIE A. (2005), « The Performativity of Code: Software and Cultures of Circulation ». *Theory Culture Society*. vol. 22, n° 1, p. 7192.

MANARA C., (2011) « Le droit d'auteur contre l'accès à l'information mondiale ? Le cas des moteurs de recherche », *Revue Internationale de Droit Économique*, vol. 25, n° 2, pp. 143-164.

MANOVICH L. (2001), *The Langage of New Media*, Cambridge, MIT Press, 2001.

MANOVICH L. (2013), *Software takes command*, New York, Bloomsbury Academic, 376 p.

MARINO M. C. (2006), « Critical code studies », *Electronic book review*.

OLIVER L. (2008a), « Google rejects adoption of ACAP standard », *Journalism.co.uk*, en ligne : http://www.journalism.co.uk/news/google-rejects-adoption-of-acap-standard/s2/a531181/

OLIVER L. (2008b), « Gavin O'Reilly responds to Google's rejection of ACAP », *Journalism.co.uk*, en ligne : http://www.journalism.co.uk/news/gavin-o-reilly-responds-to-google-s-rejection-of-acap/s2/a531184/

RIEDER B. (2006) *Métatechnologies et délégation. Pour un design orienté-société dans l'ère du Web 2.0*, thèse en Sciences de l'Information et de la Communication, Université Paris 8, Vincennes-Saint-Denis, 388 p.

ROGERS R. (2013), *Digital Methods*, Cambridge, MIT Press, 280 p.

RUSHE D. (2014), « News Corp executive labels Google a ‘platform for piracy’ », *The Guardian*, en ligne : http://www.theguardian.com/technology/2014/sep/18/google-news-corp-piracy-platform-european-commission.

SMYRNAIOS N., REBILLARD F. (2011), « Entre coopération et concurrence : Les relations entre infomédiaires et éditeurs de contenus d’actualité », *Concurrences*, n° 3.

SIRE G. (2013), *La production journalistique et Google*, thèse de doctorat en sciences de l’information et de la communication, Université Paris 2 Panthéon Assas.

SOOKMAN B. (2011), « Is Google News Legal ? », *Blog de Barry Sookman*, en ligne : http://www.barrysookman.com/2011/05/17/is-google-news-legal/

SPENCER S. (2009), « A Deeper Look At Robots.txt », *Search Engine Land*, en ligne : http://searchengineland.com/a-deeper-look-at-robotstxt-17573.

STROWEL A. (2011), *Quand Google défie le droit. Plaidoyer pour un Internet transparent et de qualité*, Bruxelles, Larcier, 240 p.

SULLIVAN D. (2007), « ACAP Launches, Robots.txt 2.0 For Blocking Search Engines ? », *Search Engine Land*, en ligne : http://searchengineland.com/acap-launches-robotstxt-20-for-blocking-search-engines-12802.

SULLIVAN D. (2009a), « Dear WSJ: To Avoid Google Disease, Please Put A Condom On Your Content », *Daggle*, en ligne : http://daggle.com/dear-wsj-avoid-google-disease-put-condom-content-1451.

SULLIVAN D. (2009b), « Head-To-Head: ACAP Versus Robots.txt For Controlling Search Engines », *Search Engine Land*, en ligne : http://searchengineland.com/head-to-head-acap-versus-robots-txt-for-controlling-search-engines-30816.

VAN ASBROECK, COCK M. (2007), « Belgian newspapers v Google News: 2-0 », *Journal of Intellectual Property Law & Practice*, vol. 2, n° 7, pp. 463-466.