



**HAL**  
open science

## Traduire la loi en code grâce au langage de programmation Catala

Liane Huttner, Denis Merigoux

### ► To cite this version:

Liane Huttner, Denis Merigoux. Traduire la loi en code grâce au langage de programmation Catala. Intelligence artificielle et finances publiques, Oct 2020, Nice, France. hal-03128248

**HAL Id: hal-03128248**

**<https://inria.hal.science/hal-03128248>**

Submitted on 2 Feb 2021

**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.

## 121 Traduire la loi en code grâce au langage de programmation Catala

**Liane Huttner**

*Doctorante à l'Université Paris-1 Panthéon-Sorbonne, IRJS-DREDIS.*

**Denis Merigoux**

*Doctorant à l'Institut national de recherche en sciences et technologies du numérique (INRIA), équipe Prosecco.*

**1 - Introduction.** - Chaque année, l'administration fiscale française doit calculer le montant imposable de plus de 38 millions de foyers fiscaux. À ce nombre déjà imposant, s'ajoutent les calculs de la TVA qui concerne près de 6 millions de sociétés, et de l'impôt sur les sociétés applicable à plus de 2 millions de sociétés. Et c'est sans compter sur les allocations familiales, le chômage, la retraite et d'autres prestations sociales. Comment, dès lors, l'État français parvient-il à effectuer tous ces calculs ? En matière fiscale, depuis les années 1980, l'administration fiscale a mis en place des algorithmes permettant de centraliser et de faciliter tous ces calculs ; les caisses d'allocation familiale ont fait de même, ainsi que les caisses de retraite et l'assurance chômage. Ces algorithmes, s'exécutant grâce à des programmes informatiques, doivent implémenter les textes de nature législative et réglementaire à qui ils correspondent. La correspondance entre les programmes et les textes est essentielle, puisque si les programmes s'écartent du texte qu'ils implémentent, leur légalité et leurs résultats peuvent être contestés. Cette divergence entre le texte de loi et le programme qui l'implémente est ce qu'on peut appeler un *bug*, c'est-à-dire un dysfonctionnement du programme. En raison de la complexité de ces programmes et de l'absence d'une réelle collaboration entre juristes et informaticiens, ces *bugs* ne sont malheureusement pas rares. Ils ont par exemple été relevés en matière de chômage<sup>1</sup> ou de salaire<sup>2</sup>. En plus de poser d'importantes questions juridiques, ces *bugs* ont des conséquences sociales significatives. C'est pourquoi il est nécessaire aujourd'hui de proposer une alternative aux solutions existantes d'implémentation informatique des lois. Le langage Catala, qui résulte d'une collaboration entre juristes et informaticiens, représente l'une de ces initiatives.

Dans cette étude, nous nous attacherons d'abord à montrer et illustrer les difficultés techniques qui se dégagent lorsqu'une loi est implémentée par un algorithme (1). Dans un second temps, nous reviendrons sur les problématiques relevant du régime juridique applicable à ces algorithmes (2). Enfin, nous présenterons le langage Catala, créé en prenant en compte ces doubles enjeux techniques et juridiques (3).

### 1. Les enjeux techniques des algorithmes d'implémentation de lois

**2 - La notion de *legacy code*.** - Écrire un programme informatique permettant d'appliquer la loi dans toute sa subtilité et le maintenir en état opérationnel pendant plusieurs dizaines d'années est un réel défi technique. Actuellement et opérationnellement, on peut remarquer que certains systèmes informatiques réalisant cette tâche sont frappés d'obsolescence. Par ailleurs, leur maintenance est coûteuse et ne permet pas d'assurer systématiquement la correction vis-à-vis de la loi. En effet, les systèmes informatiques des grandes organisations,

---

1 D. Israel, *Fraudes au chômage partiel : le contrôle impossible* : Mediapart, 26 oct. 2020.

2 P. Reltien, *Des milliers de soldats français endettés à cause de Louvois, leur logiciel de paie* : France culture, 26 janv. 2018.

dans le secteur public ou privé, datent généralement des années 1980<sup>3</sup>. C'est le cas des systèmes informatiques assurant le calcul des impôts et prestations sociales. Après plus de 30 ans d'existence et de production, ces systèmes souffrent d'une obsolescence due à l'évolution rapide de la science informatique. Concrètement, les organisations qui maintiennent ces systèmes sont victimes d'une perte de compétence, et parfois plus personne ne sait comment fonctionnent les programmes informatiques<sup>4</sup>. C'est ce que l'on appelle du *legacy code*.

**3 - Une maintenance coûteuse et partielle.** - Chaque année, la législation fiscale et sociale change et les systèmes informatiques de l'État qui en dépendent doivent être mis à jour. Cette mise à jour est rendue difficile par l'obsolescence décrite au paragraphe précédent, mais aussi par l'absence de correspondance directe entre les articles du texte de loi modifié et l'algorithme informatique effectuant le calcul qui y est décrit. C'est donc aux informaticiens responsables de la maintenance de ces systèmes de garder à jour un modèle mental de la correspondance entre articles de loi et lignes de code informatique. Ce processus est donc très vulnérable à la perte de compétence. Cette absence de correspondance impose également une validation *a posteriori* de l'algorithme, par le passage de jeux d'essai. Un jeu d'essai est un foyer fiscal ou un ménage imaginaire pour lequel des juristes ont calculé à la main le montant d'impôt ou d'allocations dus. Avant d'être déployé en production, l'algorithme doit retourner les bons résultats sur ces jeux d'essai. Bien que nécessaire, cette étape n'est pas complètement suffisante. En effet, la confiance dans la validation de l'algorithme augmente avec le nombre et la diversité des jeux d'essais sur lequel il est validé. Techniquement parlant, il faudrait plusieurs milliers de jeux d'essai pour tester l'algorithme sur toutes les configurations de ménages se présentant dans la réalité. Or, parce que la création et la mise à jour de jeux d'essai à chaque changement de législation nécessitent une expertise juridique coûteuse, les administrations n'utilisent actuellement que quelques centaines de jeux d'essais.

Cette validation insuffisante est à l'origine de nombreux *bugs* qui ont des conséquences très réelles sur les personnes soumises aux décisions de l'algorithme, allant de la surimposition au déni de droits sociaux<sup>5</sup>.

## 2. Les enjeux juridiques des algorithmes d'implémentation de lois

**4 -** Les algorithmes utilisés par l'administration fiscale pour calculer les impôts ne s'inscrivent pas dans un vide juridique. Ils doivent en effet respecter un certain nombre de règles applicables aux algorithmes administratifs (**A**), en même temps qu'ils doivent se révéler conformes aux règles de droit implémentées (**B**).

### A. - Le respect du régime juridique des algorithmes administratifs

**5 - Le régime juridique général.** - Les algorithmes utilisés par l'administration fiscale et les autres administrations obéissent à un régime juridique précis, tiré du droit des données à caractère personnel et du droit administratif. Ce droit s'est construit progressivement depuis les années 2010. Aujourd'hui, ces algorithmes doivent obéir à quatre grandes garanties pour

---

3 C'est en effet à cette époque que se développe le concept de « système expert », programme informatique dont le fonctionnement est censé suivre des règles fonctionnelles dictées par les experts du domaine sur lequel le système opère.

4 K. Leswing, *New Jersey needs volunteers who know COBOL, a 60-year-old programming language* : CNBC, 6 avr. 2020.

5 M. Goanec, *Le conseiller Pôle emploi qui aidait les allocataires vient d'être licencié* : Mediapart, 16 juill. 2020.

être en conformité avec les textes de loi : transparence du traitement algorithmique, faculté d'obtenir une intervention humaine, possibilité pour les administrés de contester la décision ainsi que maîtrise du traitement par le responsable. En matière fiscale, ce sont en particulier les garanties de transparence du traitement et de sa maîtrise qui s'avèrent problématiques. Cette étude s'attardera donc sur ces deux éléments.

**6 - Les garanties de transparence.** - La transparence des traitements algorithmiques fondant une décision est un point clé du régime des algorithmes administratifs, très discuté depuis l'affaire *Admission Post-Bac*<sup>6</sup>. À la suite de cette affaire, la loi n° 2016-1321 du 7 octobre 2016 pour une République numérique a introduit dans le Code des relations entre le public et l'administration (CRPA) une série de dispositions obligeant les administrations à communiquer les codes sources et les règles générales des algorithmes. Plus précisément, l'article L. 300-2 du CRPA, qui qualifie les codes sources produits par l'État, les collectivités territoriales ainsi que par les autres personnes de droit public ou les personnes de droit privé chargées d'une mission de service public, a pour conséquence d'obliger ces personnes à mettre à disposition ces documents dans un standard ouvert et aisément réutilisable. Pour répondre à cette obligation, le Gouvernement a mis en place un site internet ([data.gouv](http://data.gouv.fr)) permettant de réunir l'intégralité des codes sources. En plus de cette obligation générale de publication, les articles L. 311-3 et suivants obligent les administrations à mentionner l'utilisation d'algorithmes lors de la prise de décision. Les citoyens bénéficient par ailleurs d'un droit d'accès à ces traitements lorsqu'ils en formulent la demande. La transparence des algorithmes administratifs est donc une obligation développée et acquise dans l'ordre juridique français. Mais paradoxalement, ces obligations ne sont pas toujours bien remplies. Si de nombreuses administrations publient bel et bien leurs codes sources, l'accès des citoyens aux règles définissant les traitements algorithmes n'est pas toujours aisé<sup>7</sup>. La Commission d'accès aux documents administratifs (Cada) traite ainsi de nombreuses demandes faites par les justiciables<sup>8</sup>. En 2019, elle a une nouvelle fois incité la DGFIP à publier les règles algorithmiques du calcul de l'impôt sur le revenu<sup>9</sup>. Les demandes des contribuables sont coûteuses et longues à traiter. Elles traînent et sont parfois impossibles à traiter d'autant plus que les systèmes reposent sur des technologies obsolètes et propriétaires impossibles à répliquer en dehors de l'administration, à l'exact inverse de la culture de l'*open source* qui se développe depuis les années 2000<sup>10</sup>. À ces premières difficultés pour répondre aux exigences de transparence, s'ajoutent celles liées à la maîtrise du traitement.

**7 - La maîtrise du traitement algorithmique.** - L'exigence de maîtrise du traitement algorithmique fondant une décision est une obligation prévue par l'article 47, § 2, 2° de la loi

---

6 V. J. Rochfeld, *Droit des personnes. Droit de ne pas subir une décision fondée exclusivement sur un traitement automatisé*, in *Droit des données personnelles, Les spécificités du droit français au regard du RGPD* : Dalloz, coll. *Décryptage*, 2019, p. 175. - *Rép. IP/IT et Communication Dalloz*, V° *Données à caractère personnel. Droit de ne pas subir une décision fondée sur un traitement automatisé*, n° 17.

7 D. Merigoux, R. Monat et C. Gaie, *Étude formelle de l'implémentation du code des impôts*, 31<sup>e</sup> Journées Francophones des Langages Applicatifs, 2020, disponible sur <https://hal.inria.fr/hal-02320347>.

8 Plus de deux cents demandes ont été traitées par la Cada depuis 2017 concernant la publication de codes sources et de règles des traitements algorithmiques. Ces demandes portent autant sur l'impôt sur le revenu que la taxe d'habitation, l'impôt sur les sociétés et la TVA. Pour plus d'informations, le site de la Cada regroupe tous les avis : [https://cada.data.gouv.fr/search?administration=Direction + g % C3 % A9n % C3 % A9rale + des + finances + publiques + % 28DGFIP % 29](https://cada.data.gouv.fr/search?administration=Direction+g%C3%A9n%C3%A9rale+des+finances+publiques+%28DGFIP%29).

9 Cada, avis n° 20185134, 6 juin 2019 : publication en ligne des « règles » définissant les principaux traitements algorithmiques utilisés dans l'accomplissement des missions de la DGFIP « lorsqu'ils fondent des décisions individuelles », tel que le prévoit l'article L. 312-1-3 du Code des relations entre le public et l'administration.

10 V. par ex., Cada, avis n° 20181891, 18 juill. 2019 : [www.cada.fr/20181891](http://www.cada.fr/20181891).

n° 78-17 du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés, dite « informatique et libertés », telle que modifiée par la loi n° 2018-493 du 20 juin 2018 relative à la protection des données personnelles. Plus précisément, l'article dispose qu'en ce qui concerne les décisions administratives, « *le responsable de traitement s'assure de la maîtrise du traitement algorithmique et de ses évolutions afin de pouvoir expliquer, en détail et sous une forme intelligible, à la personne concernée la manière dont le traitement a été mis en œuvre à son égard* ». Dans sa décision n° 2018-765 DC du 12 juin 2018 portant sur la loi relative à la protection des données personnelles <sup>11</sup>, le Conseil constitutionnel a insisté sur cette garantie en obligeant les responsables de traitement à s'assurer de la maîtrise du traitement algorithmique et de ses évolutions, excluant par-là l'utilisation d'algorithmes auto-apprenants <sup>12</sup>. Mais la maîtrise du traitement algorithmique s'étend au-delà de l'interdiction d'utiliser des algorithmes auto-apprenants. En effet, cette obligation négative pourrait s'accompagner d'une obligation positive de réellement maîtriser le traitement, c'est-à-dire de le comprendre, de le contrôler et de bien le connaître. Une réelle maîtrise du traitement algorithmique signifie que les responsables comprennent systématiquement comment les résultats ont été obtenus. Or, l'empilement des règles algorithmiques correspondant à l'inflation législative depuis les années 1980, conjuguées à l'impossibilité pour des organisations comme Pôle Emploi, la CNAF ou la CNAV de produire systématiquement une explication détaillée et individuelle du montant calculé pour les pensions ou allocations font douter d'une réelle maîtrise du traitement. Sur ce point, la DGFIP s'en tire mieux grâce à son langage de programmation dédié « M », mais cette technologie montre ses limites et est-elle aussi frappée par la perte de compétence. Ces doutes sur la maîtrise du traitement algorithmique, ajoutés au coût des obligations de transparence, rendent les algorithmes utilisés par l'administration fiscale largement contestables. Or, ces enjeux ne constituent qu'une partie du problème. La question du respect de la loi fiscale par les algorithmes d'implémentation rend la situation encore plus fragile.

## **B. - Le respect de la norme juridique implémentée**

**8 - Les deux catégories d'algorithmes administratifs.** - Alors que les règles de transparence, d'intervention humaine et de maîtrise du traitement algorithmique sont relativement clarifiées, la question du respect par les algorithmes des textes implémentés est beaucoup plus complexe. Pour bien comprendre la question, il convient tout d'abord de revenir sur la différence entre les algorithmes d'implémentation de normes juridiques, et ceux qui possèdent une relative autonomie par rapport à de tels textes. Dans le premier cas, qui nous intéresse ici, l'algorithme n'a pas pour objet de modifier l'ordonnancement juridique. Il n'est conçu que pour faciliter l'action de l'administration, en suivant le plus exactement possible le texte implémenté. Bien souvent, l'algorithme est d'ailleurs conçu uniquement comme une modalité de mise en œuvre. La situation est très différente dans le cas où l'administration crée un algorithme non fondé sur un texte de loi. Ces algorithmes peuvent créer de nouveaux référentiels (tel est le cas par exemple du traitement DataJust conçu pour élaborer un référentiel indicatif facilitant l'indemnisation des préjudices corporels) <sup>13</sup>, ou alors traiter de grandes bases de données à caractère personnel (comme les algorithmes de

---

11 *Cons. const., 12 juin 2018, n° 2018-765 DC sur la loi relative à la protection des données personnelles : Comm. com. électr. 2018, comm. 65, obs. A. Debet.*

12 *Cons. const., 12 juin 2018, n° 2018-765 DC sur la loi relative à la protection des données personnelles, préc., pt 71.*

13 *D. n° 2020-356, 27 mars 2020, portant création d'un traitement automatisé de données à caractère personnel dénommé « DataJust ».*

traitement des données passagers) <sup>14</sup>. S'ils doivent respecter les règles générales applicables aux algorithmes administratifs, ils ne doivent pas être strictement conformes à un texte législatif ou réglementaire, dont ils sont issus.

**9 - L'obligation juridique de respecter le texte implémenté.** - Les algorithmes d'implémentation de normes juridiques doivent donc, par définition, respecter strictement les textes dont ils sont issus. Pourtant, leur situation juridique n'est pas si simple et contraste avec cette règle de bon sens. Plus précisément, la question posée est la suivante : si un algorithme ne respecte pas la norme juridique qu'il implémente, est-il valable ? Peut-il être utilisé par l'administration ? Où peut-il être contesté, et le cas échéant, annulé ? Pour répondre à ces questions, deux hypothèses peuvent se présenter. En premier lieu, on pourrait imaginer qu'une règle de droit oblige les concepteurs d'algorithmes d'implémentation de loi à respecter strictement la norme juridique implémentée. Cette règle de droit pourrait en même temps déterminer les marges de manœuvre qu'ont les concepteurs des algorithmes. Cela signifierait que tout algorithme ne respectant pas strictement le texte de loi et dépassant les marges de manœuvre expresses pourrait être contesté. La situation aurait le mérite d'être claire, et les obligations de chacun seraient explicites. Pourtant, à notre connaissance, il n'existe aucun texte de droit positif qui règle la question. Que ce soit en droit fiscal ou en droit social, les administrations n'ont pas d'obligation positive de respecter les textes de loi implémentés. Si une telle absence est à déplorer en ce qu'elle permettrait de clarifier la situation, cela signifie aussi donc qu'il est nécessaire de trouver d'autres fondements de contestation des algorithmes administratifs. Nous avons vu que ces algorithmes étaient soumis à un régime particulier des documents administratifs - c'est donc leur première qualification. Mais une seconde qualification pourrait leur être attribuée et avoir d'importantes conséquences : si les algorithmes administratifs pouvaient être qualifiés d'actes administratifs, cela signifierait qu'ils pourraient faire l'objet d'un recours pour excès de pouvoir, sanctionnant leur non-conformité aux normes juridiques qu'ils doivent implémenter.

**10 - La possibilité d'un recours en excès de pouvoir contre un algorithme administratif.** - Aucun algorithme administratif n'a, à notre connaissance, fait l'objet d'un recours en excès de pouvoir. Mais une telle possibilité n'est pas à exclure. Pour que ce soit le cas, l'algorithme devrait répondre à un trois critères : il doit être qualifié d'acte administratif, d'acte unilatéral et d'acte faisant grief. Les deux premières conditions ne posent pas véritablement de problème. Le caractère administratif de l'acte tient de son auteur et de son objet <sup>15</sup>. Si ce caractère a pu poser des difficultés concernant certains types d'actes, la qualité administrative de l'auteur de l'acte permet de présumer le caractère administratif de l'acte. Un algorithme conçu par l'administration fiscale a donc bien un caractère administratif. De la même manière, un acte unilatéral est destiné à régir le comportement des personnes étrangères à son édicton <sup>16</sup>. Les obligations issues de cet acte s'imposent donc aux destinataires sans leur consentement. Ce critère est bien rempli pour les algorithmes de calcul des impôts, dont le résultat est imposé aux contribuables. En revanche, la question se complexifie lorsqu'on s'interroge sur la qualification de l'algorithme administratif comme un acte faisant grief. De manière générale, le Conseil d'État requiert que l'acte lèse le requérant <sup>17</sup>, c'est-à-dire qu'il porte atteinte à sa

---

14 V. not., *PE et Cons. UE, dir. 2016/681/UE, 27 avr. 2016, relative à l'utilisation des données des dossiers passagers (PNR) pour la prévention et la détection des infractions terroristes et des formes graves de criminalité, ainsi que pour les enquêtes et les poursuites en la matière.*

15 V. *JCl. Administratif, fasc. 1140, Conditions d'exercice du recours pour excès de pouvoir. Conditions particulières relatives à la nature de l'acte contesté, 2011, n° 76, par A. Perrin.*

16 V. *JCl. Administratif, fasc. 1140, Conditions d'exercice du recours pour excès de pouvoir. Conditions particulières relatives à la nature de l'acte contesté, 2011, n° 120, par A. Perrin.*

17 *CE, 9 mars 1938, Janes : Lebon, p. 254.*

situation juridique ou matérielle. L'acte faisant grief doit donc produire des effets sur l'ordonnancement juridique<sup>18</sup>. Une telle qualification a été reconnue à des actes aussi divers que les circulaires à caractère impératif<sup>19</sup>, à des prises de position de l'administration<sup>20</sup> et même à des décisions par lesquelles l'administration décide de procéder à la publication de documents émanant de tiers<sup>21</sup>. En ce qui concerne les algorithmes administratifs calculant des impôts, on peut avancer l'hypothèse selon laquelle ils modifient bien l'ordonnancement juridique : c'est sur leur fondement que l'impôt des contribuables va être calculé. Ils affecteront donc la situation du requérant, en particulier si certaines erreurs modifient la somme imposée. Si ces trois caractères (administratif, unilatéral et faisant grief) sont reconnus, il est donc parfaitement possible que les algorithmes servant à calculer les impôts puissent faire l'objet d'un recours en excès de pouvoir.

**11 - Les conséquences d'un recours en excès de pouvoir contre un algorithme administratif.** - Si on admet la possibilité d'un recours en excès de pouvoir contre un algorithme administratif, plusieurs possibilités d'annulation se présentent. Le fondement le plus intéressant à notre sens est celui de la violation directe de la loi. En effet, l'algorithme de calcul des impôts est une modalité d'application de la loi fiscale. Toute erreur d'implémentation et de modélisation violerait donc immédiatement la loi, en ayant des conséquences significatives sur la somme imposée. Ces erreurs constitueraient une violation directe de la règle de droit. Le contrôle par le juge de la violation directe de la loi est souvent un contrôle de non-contrariété à la norme supérieure, ce qui signifie que le juge doit vérifier que l'administration n'ait rien fait qui soit en contradiction avec le texte. Une erreur dans l'algorithme de calcul des impôts pourrait être en violation de la loi, à partir du moment où cette erreur s'oppose aux termes législatifs ou réglementaires. Un autre fondement d'annulation pourrait être constitué par l'erreur sur le sens et la portée de la règle de droit. Le juge administratif peut en effet annuler des décisions qui manifestent une interprétation inexacte de la règle de droit<sup>22</sup>. Si une interprétation inexacte de la loi fiscale se retrouve dans l'algorithme de calcul de l'imposition, cela pourrait donc constituer un fondement d'annulation.

**12 - Une sécurité juridique insuffisante.** - Si les algorithmes de calcul des impôts ne respectent pas toutes les obligations décrites ci-dessus, la question de la sécurité juridique se pose sérieusement. Tant la maîtrise du traitement que la conformité de l'algorithme à la loi fiscale peuvent être contestées. Cette contestation pourrait entraîner l'annulation de ces algorithmes administratifs. L'annulation de ces algorithmes pourrait mener l'administration à devoir construire de nouveaux algorithmes de calcul des impôts en conformité avec les textes de loi et l'interprétation fiscale en peu de temps et produirait des coûts élevés. De la même manière, l'administration fiscale se voit confrontée à de nombreuses demandes de publications qui entraînent des coûts et demandent du temps de traitement. Pour toutes ces raisons, il est

---

18 V. JCl. Administratif, fasc. 1140, *Conditions d'exercice du recours pour excès de pouvoir. Conditions particulières relatives à la nature de l'acte contesté*, 2011, n° 13, spéc. n° 27, par A. Perrin.

19 CE, sect., 18 déc. 2002, n° 233618, Duvignères : *JurisData* n° 2002-064827 ; Lebon, p. 463 ; *Procédures* 2003, n° 154, note S. Deygas ; *Dr. adm.* 2003, comm. 73 ; Lebon, p. 463 ; *RJS* 2003, n° 688 ; *RFD adm. mars-avr. 2003, concl. P. Fombeur*, p. 280 ; *AJDA* 17 mars 2003, *chron. F. Donnat et D. Casas*, p. 487.

20 CE, ass., 8 avr. 2009, n° 311136, Hollande et Mathus : *JurisData* n° 2009-075240 ; Lebon, p. 140 ; *RFD adm.* 2009, p. 35, note C. de Salins.

21 CE, Ire et 6e ss-sect., 16 janv. 2006, n° 274721 et n° 274722 : *JurisData* n° 2006-069516 ; *Dr. adm.* 2006, comm. 48 ; *Contrats, conc. consom.* 2006, comm. 117, G. Raymond ; *Gaz. Pal.* 15 août 2006, p. 17, note Graveleau ; *D.* 2006, p. 576, obs. Avena-Robardet.

22 V. JCl. Administratif, fasc. 1152, *Recours pour excès de pouvoir. Contrôle de la légalité interne*, 2009, n° 99, par D. Péano.

nécessaire que l'administration et notamment l'administration fiscale utilisent des outils plus adaptés.

### 3. Catala comme solution aux enjeux techniques et juridiques des algorithmes d'implémentation de lois

**13 - La conformité aux obligations de transparence grâce au développement en *open source*.** - Pour répondre à ces enjeux, plusieurs alternatives existent, et en particulier le langage Catala. Catala est un langage informatique créé par une équipe de juristes et d'informaticiens qui a exactement pour objectif de répondre aux enjeux développés dans cet article, grâce à trois points essentiels. En premier lieu, Catala est développé en *open source*. Cela signifie que le code informatique de l'application ainsi que son historique de modification sont disponibles en ligne et accessibles à tous, à l'instar d'un texte législatif sur Légifrance. Conjuguée à l'utilisation d'une infrastructure d'exécution, elle aussi en *open source* et répliquable en dehors de l'administration, cette mise à disposition du code entraîne une transparence *a priori* de l'algorithme. Cette transparence est également pensée dans le temps long, puisqu'un dispositif similaire à celui utilisé sur Légifrance permet de visualiser les modifications apportées au code source, et permet ainsi un audit incrémental par les citoyens.

**14 - La maîtrise du traitement algorithmique grâce aux méthodes formelles.** - Une seconde innovation de Catala consiste en l'utilisation des méthodes formelles. Les méthodes formelles sont une branche de la science informatique qui considère les algorithmes comme des objets mathématiques sur lesquels il est possible de raisonner et de prouver des théorèmes. Les méthodes formelles sont utilisées industriellement dans les systèmes informatiques critiques (fusées, avions, trains, centrales nucléaires) pour éliminer à l'avance, et de manière certaine, des catégories entières de bugs et augmenter la sécurité du logiciel. Appliquer les méthodes formelles aux algorithmes de traitement administratif permet d'éliminer à l'avance toute possibilité de *bug* contingent au détail bas niveau du système informatique, et de se concentrer sur la correction législative du code en lui-même.

**15 - La conformité aux textes de loi grâce à la programmation littérale en binôme.** - Enfin, la programmation en Catala s'effectue de manière littérale, mettant en jeu un binôme d'informaticien et de juriste. La programmation littérale est un style d'écriture des programmes informatiques où chaque ligne de code est annotée d'une ligne de texte décrivant son sens et son objectif en bon français. Appliqué à notre situation, cela veut dire que le programme peut être directement entremêlé localement au texte de loi, chaque article et chaque alinéa possédant sa traduction locale en termes algorithmiques. La conformité du langage informatique au texte de loi implémenté pourra donc être systématiquement et localement vérifiée, puisque l'implémentation sera reliée au texte de loi original. Par ailleurs, l'implémentation sera faite par des binômes constitués d'un juriste et d'un informaticien. Catala ayant été développé dans ce cadre, le langage peut être compris autant par des juristes que par des informaticiens, qui collaboreront lors de la programmation. Toute question de cohérence ou d'interprétation pourra ainsi être résolue grâce aux compétences des deux parties. Cette méthode de travail évitera que des informaticiens interprètent la loi dans le savoir en prenant des décisions unilatérales dans leur code informatique, mais aussi que les juristes se rendent compte d'une ambiguïté du texte soulevée par la traduction algorithmique.

**16 - Conclusion.** - Angle mort de la réflexion actuelle sur les algorithmes, la correction des systèmes informatique existants utilisés par les administrations est un problème important. Les difficultés techniques et le manque de ressources alloués à la validation juridique de ces



programmes pourraient entraîner l'annulation pure et simple de l'algorithme utilisé. Pour répondre à cet enjeu, Catala fait ainsi partie d'une nouvelle génération de langages informatiques conçus en prenant en compte toutes les difficultés présentées. Son utilisation permettrait de garantir avec efficacité et à relativement faible coût la conformité au droit des données personnelles et au droit administratif, tout en obéissant à un principe de transparence.

**Mots-clés : Intelligence artificielle et finances publiques - Langage Catala**