

*Warren Sack*

UNE MACHINE A RACONTER DES HISTOIRES :  
DE PROPP AUX *SOFTWARE STUDIES*

Dans son célèbre article sur l'analyse structurale des mythes, Claude Lévi-Strauss décrivait sa démarche en ces termes :

« Nous avons, jusqu'à présent, utilisé la technique suivante : chaque mythe est analysé indépendamment, en cherchant à traduire la succession des événements au moyen des phrases les plus courtes possibles. Chaque phrase est inscrite sur une fiche qui porte un numéro correspondant à sa place dans le récit<sup>1</sup>. »

Il arrange ensuite ces fiches sur une table en rangées et en colonnes. Chaque colonne correspond à une fonction ou à la permutation d'une fonction. Chaque rangée peut être lue de gauche à droite, puisqu'elle comprend une série d'une ou plusieurs actions en séquence chronologique. Le résultat est une vaste grille de fiches pour une variante d'un mythe donné. Pour comparer les variantes, il fait de même pour chacune d'entre elles, essayant d'égaliser le nombre de colonnes et de fonctions qui leur correspondent, de sorte que la comparaison entre les variantes d'un mythe est une analyse compliquée de ce qui est essentiellement un tableau en trois dimensions. Lévi-Strauss commente ainsi son travail :

« Les textes mythiques sont extrêmement volumineux. Leur analyse en unités constitutives exige un travail d'équipe et un personnel technique. Une variante de dimension moyenne fournit

---

1. Claude Lévi-Strauss, *Anthropologie structurale*, Paris, Plon, 1958, p. 233.

plusieurs centaines de cartes. Pour découvrir la meilleure disposition de ces cartes en colonnes et en rangées, il faudrait avoir des classeurs verticaux d'environ 2 m × 1,50 m, garnis de casiers où l'on puisse répartir et déplacer les cartes à volonté. Et dès qu'on propose d'élaborer des modèles à trois dimensions, pour comparer plusieurs variantes, autant de classeurs que de variantes sont nécessaires, ainsi qu'un espace suffisant pour les mouvoir et les disposer librement. Enfin, si le système de référence fait appel à plus de trois dimensions (ce qui risque de se produire rapidement, comme on l'a montré à la p. 242), il faut recourir aux cartes perforées et à la mécanographie<sup>2</sup>. »

Comprendre une histoire, dans les termes de Lévi-Strauss, équivaut à la répartir en un tableau en trois dimensions et à définir un ensemble de relations entre les rangées et les colonnes de ces tableaux. Comme Michel Foucault l'a écrit, la théorie sociale et l'ordre social dépendent depuis longtemps de tableaux<sup>3</sup>. Toutefois, ce qui est remarquable dans ce court passage est le fait que, pour trouver un support matériel adéquat à ses besoins, Lévi-Strauss suggère qu'il va falloir renoncer à la forme imprimée, pour se tourner vers un traitement informatique. Sans réduire le mérite de Lévi-Strauss, on peut considérer son entreprise comme l'illustration d'une loi plus générale qui veut que ce qui compte comme compréhension, dans n'importe quelle période historique donnée, est la capacité d'articuler les formes plus anciennes de connaissance dans les tropes de la technologie la plus récente. Comprendre, dans une société où l'écriture est la technologie la plus récente, demanderait qu'on mette sur papier les histoires orales<sup>4</sup>. Comprendre, dans une société dominée par l'imprimerie, consisterait à appliquer les techniques de l'imprimerie aux textes créés par les scribes, donc introduire des chapitres, des notes de bas de page, une table des matières, un index par matières, et tout l'appareil de la bibliothèque. Comprendre, dans une société dominée par l'informatique, consisterait à transcoder les textes imprimés en logiciels, comme

2. *Ibid.*, p. 253. Il est à noter ici que le mot « mécanographie » est exprimé dans la version originale anglaise (car le texte a d'abord paru en anglais) par l'expression « I.B.M. equipment », autrement dit par des ressources informatiques.

3. Michel Foucault, *Les Mots et les Choses*, Paris, Gallimard, 1966.

4. Cf. Eric Alfred Havelock, *Preface to Plato*, Cambridge, Belknap Press Harvard University Press, 1963.

l'avait d'ailleurs prédit Jean-François Lyotard<sup>5</sup>. Comprendre une société connectée par l'Internet serait la décrire en utilisant les technologies du Web 2.0, non comme objets d'analyse empirique, mais comme les lentilles théoriques par lesquelles il faut examiner la dynamique sociale de l'Internet elle-même<sup>6</sup>. Si nous suivons ce raisonnement, la base de données et le récit ne seraient pas des formes opposées, contrairement à ce qu'a écrit Lev Manovich<sup>7</sup>; les bases de données seraient simplement une composante indispensable à la compréhension contemporaine des récits. Le but de cet article est de montrer que l'on peut reprendre l'analyse par Vladimir Propp des contes merveilleux russes pour les réinscrire dans la technologie la plus récente, c'est-à-dire dans les termes d'un logiciel. Il ne s'agit pas de montrer que le formalisme en littérature a été un moment dans cette vaste réécriture numérique du monde qui est, pour le meilleur ou pour le pire, notre présent et notre avenir. Il s'agit plutôt de définir le lieu où les sciences de l'informatique et les humanités peuvent instaurer un commerce intelligent et profitable à tous.

---

5. « L'orientation des recherches nouvelles se subordonnera à la condition de traduisibilité des résultats éventuels en langage de machine. Les "producteurs" de savoir comme ses utilisateurs doivent et devront avoir les moyens de traduire dans ces langages ce qu'ils cherchent les uns à inventer, les autres à apprendre. » (Jean-François Lyotard, *La Condition postmoderne : rapport sur le savoir*, Paris, Minuit, 1979, p. 13.)

6. Cf., Richard Rogers, « The End of the Virtual : Digital Methods » (Vossiuspers UvA, 2009), et Latour, « The Whole Is Always Smaller Than Its Parts : A Digital Test of Gabriel Tarde's Monads ».

7. Dans son article « Database as Symbolic Form » (La base de données en tant que forme symbolique : chapitre 5 de son livre *The Language of New Media*), Lev Manovich compare les formes des bases de données avec celles du récit et conclut : « En tant que forme culturelle, la base de données représente le monde comme une liste d'éléments et elle refuse de mettre cette liste en ordre. A l'opposé, un récit crée une trajectoire cause-effet composée d'éléments (d'événements) apparemment désordonnés. Donc, la base de données et le récit sont naturellement opposés. Faisant la concurrence à l'autre pour le même territoire de la culture humaine, chacun d'entre eux prétend avoir le droit exclusif de faire, du monde, le sens. » (Lev Manovich, *The Language of New Media*, Cambridge, Mass, MIT Press, 2002.)

DE LA NARRATOLOGIE À L'INFORMATIQUE :  
PROPP, LÉVI-STRAUSS ET LES ORDINATEURS

Quand Vladimir Propp écrit *La Morphologie du conte* en 1928, il se propose de prendre en compte les « relations » plutôt que les « atomes » des contes. Il écrit : « Quelles méthodes peuvent accomplir une description fidèle du conte ? Comparons les événements suivants : 1) un tsar donne un aigle à l'un de ses braves. L'aigle emporte le brave dans un autre royaume. 2) son grand-père donne un cheval à Soutchenko. Le cheval emporte Soutchenko dans un autre royaume. 3) Un magicien donne un petit bateau à Ivan. Le bateau emporte Ivan dans un autre royaume, et ainsi de suite. Ces différents exemples présentent des grandeurs constantes et des grandeurs variables. Les noms des protagonistes (et leurs attributs) varient, mais leurs actions, ou leurs fonctions, ne varient pas. D'où l'on peut conclure que, souvent, les contes prêtent des actions identiques à des personnages différents. Cela nous donne la possibilité d'étudier les contes d'après les fonctions des personnages<sup>8</sup>. »

L'analyse « fonctionnelle » des contes par Propp aboutit à ces conclusions surprenantes : a) il n'y a qu'un nombre limité de fonctions (selon lui trente-et-une); b) la séquence des fonctions est toujours la même, quel que soit le conte; et par conséquent, c) tous les contes de fées ont la même structure<sup>9</sup>. Notez que les fonctions de Propp ne sont pas de simples actions; elles sont des actions en contexte. Donc, ce n'est pas assez de savoir qu'un personnage en a tué un autre pour déterminer la fonction narrative de cet acte. Un meurtre pourrait être un acte de bassesse ou d'héroïsme selon sa position dans la série de fonctions qui constituent l'histoire.

La séquence de Propp commence avec ces trois fonctions : (i) éloignement : un des membres d'une famille s'absente du foyer; (ii) interdiction : une interdiction est donnée au héros; (iii) transgression de l'interdit : l'interdiction est transgressée<sup>10</sup>. A la fin de la séquence de fonctions se trouvent celles-ci; (xxx) punition : le vilain est puni; (xxxi) mariage : le héros se marie et monte sur le

8. Vladimir Propp, *Morphology of the Folktale*, Austin, University of Texas Press, 1968, pp. 19-20.

9. *Ibid.*, pp. 21-23 et p. 64.

10. Propp, *Morphology of the Folktale*, pp. 26-27.

trône<sup>11</sup>. Entre-temps, le héros fait face à une série d'épreuves, aidé et défié par d'autres personnages.<sup>12</sup>

Lévi-Strauss a notoirement critiqué la méthode d'analyse des mythes de Propp, arguant que « [l]es véritables unités constitutives du mythe ne sont pas les relations isolées, mais des *paquets de relations*<sup>13</sup>... » Il trouve inadéquate la série de trente-et-une fonctions de Propp dans la mesure où elle n'a pas réussi à prendre en compte les ressemblances entre fonctions. Lévi-Strauss affirme que, tout comme en mathématiques une nouvelle fonction peut être composée en utilisant une fonction donnée, les fonctions narratives peuvent être identifiées comme permutations ou compositions d'autres fonctions narratives. Par exemple, en mathématiques, la variable  $f(x)$  étant donnée, l'on peut composer son inverse,  $1/f(x)$ , ou sa négation  $-f(x)$ . Combinant les permutations de fonctions, on peut identifier les « agrégats de relations » que Lévi-Strauss espérait encoder grâce à un équipement informatique<sup>14</sup>. Par exemple, on pourrait montrer que le départ du héros et son retour sont des fonctions inverses.

11. *Ibid.*, p. 63.

12. Pour Propp, les catégories des personnages doivent être définies par les fonctions qu'ils exécutent et non par leurs apparences ou leurs attributs. En plus des trente-et-une fonctions, Propp note sept « sphères d'action » (*ibid.*, p. 79), auxquelles correspond une catégorie de personnage. Les héros, par exemple, partent en quête au début d'un conte et se marient à la fin du conte. Avec les héros, Propp identifie les vilains, les donateurs, les auxiliaires, les princesses, les mandateurs, et les faux héros. Les personnages peuvent jouer un ou plusieurs rôles, et un ou plusieurs personnages peuvent jouer un rôle. Les développements ultérieurs de la théorie narrative ont élaboré et généralisé ces rôles pour les adapter à d'autres éléments d'histoires. C'est ainsi qu'en 1966 Algirdas Julien Greimas a poussé plus loin les fonctions et les « sphères d'action » de Propp pour en venir à ce qu'il appelait un « modèle actantiel ». (Cf. Algirdas Julien Greimas, *On Meaning : Selected Writings in Semiotic Theory*, Minneapolis, University of Minnesota Press, 1987, « Actants, Actors, and Figures », pp. 106-120.) Ces développements sont toujours avec nous. Par exemple, Bruno Latour et ses collègues ont développé la théorie de l'acteur-réseau basée dans les « actants » de Greimas. La TAR [ANT en anglais] a beaucoup été utilisée pour analyser les récits de la science et de la technologie e.g., John Hassard et John Law, *Actor Network Theory and After*, Oxford [England]; Malden, MA, Blackwell/Sociological Review, 1999.

13. Lévi-Strauss, *Anthropologie structurale*, pp. 233-234.

14. « Structure and Form : Reflections on a work by Vladimir

La discussion de Propp par Lévi-Strauss emploie une notation pour les fonctions que nous connaissons tous grâce à l'algèbre élémentaire enseignée à l'école. Mais, dans la plupart des cours d'algèbre, des variables numériques sont supposées prendre la place de lettres et les fonctions définissent une relation entre des ensembles de chiffres<sup>15</sup>. Or les fonctions de Propp ne sont pas définies sur des chiffres mais sur des textes. C'est la raison pour laquelle les commentaires de Lévi-Strauss peuvent à bon droit apparaître finalement comme métaphoriques plus que techniques. Plus précisément, la compréhension de Propp par Lévi-Strauss était exprimée au moyen d'une algèbre que nous reconnaissons aujourd'hui comme désuète parce qu'elle s'applique seulement aux chiffres et non aux textes.

Dans son livre, Propp attribue une lettre à chacune de ses trente-et-une fonctions et utilise des exposants et des symboles spéciaux pour indiquer les variations d'une fonction. Il déclare que les 450 contes qu'il a examinés ont une seule morphologie qui peut être décrite en utilisant ce schéma ou cette formule<sup>16</sup> :

$$\begin{array}{c} \text{HJK} \downarrow \text{PR-RS}^\circ \text{L} \\ \text{ABC} \uparrow \text{DEFG} \text{-----} \text{QEXTUW}^* \\ \text{LMJNK} \downarrow \text{PR-RS} \end{array}$$

Ceci décrit la forme générale d'une histoire dans laquelle la fonction A précède la fonction B, qui précède la fonction C, etc. La ligne horizontale au milieu n'est pas un opérateur de division, comme l'algèbre ordinaire pourrait nous le faire croire. Elle désigne un point d'embranchement après la fonction G qui pourrait, alternativement, être suivie par la fonction H ou la fonction L. Les deux branches se réunissent alors à la fonction Q<sup>17</sup>.

Propp », dans V. I. A. Liberman Anatoly Propp, *Theory and History of Folklore*, Minneapolis, University of Minnesota Press, 1984, pp. 183-184.

15. La fonction successeur  $f(x) = x + 1$  est un exemple d'une telle fonction. Il est facile de combiner des fonctions définies sur des chiffres avec des opérateurs arithmétiques comme +, -, et/. Donc, si  $g(x) = x + 2$ , une définition comme celle-ci :  $h(x) = f(x) + g(x)$  est parfaitement claire. Voir la page Wikipedia sur le Lambda-calcul : <https://fr.wikipedia.org/wiki/Lambda-calcul>

16. Propp, *Morphology of the Folktale*, p. 105.

17. Pour un résumé, voir Frederic Jameson, *The Prison-House of*

A voir la formule de Propp, on comprend pourquoi Lévi-Strauss a pu être tenté de la représenter comme une équation algébrique.

Un détail obscur, parmi d'autres, de l'exposition de Propp, est la question du sens exact de la position d'une fonction à côté d'une autre, par exemple dans la formule « AB ». Il est clair que la juxtaposition indique une séquence chronologique (A vient avant B). Mais il est également clair qu'aucune de ces fonctions ne peut être entièrement indépendante de l'autre si elles sont supposées décrire un récit cohérent qui lie une partie de l'histoire à une autre. Ainsi, dans la notation algébrique, chacune des fonctions pourrait avoir, au moins, un paramètre pour chaque membre du drame (les héros, les vilains, les princesses, etc.), afin que, par exemple, le héros du début de l'histoire ne soit pas oublié. Une telle suggestion pourrait impliquer une notation comme celle-ci : « A (héros, méchant, princesse) B (héros, méchant, princesse), etc. » Cependant, même dans ce cas, cette notation pourrait faire perdre beaucoup d'information importante. Et si, à mi-chemin de l'histoire, le héros devenait un méchant ? Il y a beaucoup d'informations implicites dans chaque fonction et cette information dépend de ce qui est déjà arrivé dans l'histoire avant la fonction. Bref, nous avons besoin d'une meilleure notation qui permet le codage du contexte narratif de chaque fonction.

Le livre de Propp, d'abord publié en russe en 1928, n'a pas été traduit en anglais avant 1958. Mais très vite, dès les années 1960, linguistes et folkloristes ont perçu le lien avec les propositions (alors nouvelles) de l'informatique. Ainsi, en 1965, le folkloriste Alan Dundes a prétendu avoir programmé *La Morphologie* de Propp pour un ordinateur<sup>18</sup>.

George Lakoff a écrit en 1964, alors qu'il n'était encore que doctorant en linguistique, une analyse remarquable de l'œuvre de Propp<sup>19</sup> qui utilise le formalisme computationnel de la linguistique,

*Language; a Critical Account of Structuralism and Russian Formalism*, Princeton, N.J., Princeton University Press, 1972, pp. 65-66.

18. Alan Dundes, « On Computers and Folktales », *Western Folklore*, 24 (1965).

19. George Lakoff, « Structural Complexity in Fairy Tales », *The Study of Man* 1 (1972). On peut trouver curieux que la grammaire transformationnelle de Chomsky ait semblé un outil approprié à l'approche de Propp. Il faut cependant rappeler que Roman Jakobson a été à la fois le professeur de Noam Chomsky et un formaliste russe. Voir

récemment introduit par Noam Chomsky : la grammaire transformationnelle<sup>20</sup>. Lakoff soutient que le formalisme de Propp est inadapté parce qu'il n'a pas les moyens de représenter le contexte. Celui-ci peut, en revanche, être saisi par les formalismes de la grammaire transformationnelle. Il réécrit certains aspects fondamentaux de l'œuvre de Propp sous la forme d'une grammaire transformationnelle.

Selon Lakoff, « les transformations sont des combinaisons de permutations, additions, oblitérations et substitutions<sup>21</sup> ». Autrement dit, pour introduire un contexte et donc une cohérence entre les fonctions de Propp, on a besoin des moyens suivants : la permutation (pour réordonner leur séquence); la substitution (pour réécrire des fonctions abstraites dans des termes plus précis qui seront, en fin de compte, restitués en français, ou dans quelque autre langue naturelle); enfin, les additions et les oblitérations (pour ajouter ou supprimer des assertions d'une base de données qui enregistre l'état courant de l'histoire); ainsi, l'assertion qu'un personnage périt au début du conte doit être enregistré pour que ce même personnage n'apparaisse pas vivant plus tard dans le conte.

Sheldon Klein était un professeur d'informatique à l'université du Wisconsin. Il avait travaillé avec Lévi-Strauss à Paris à l'École des hautes études en sciences sociales au cours de l'année 1976-1977<sup>22</sup>. Au début des années 1960, Klein et ses étudiants produisirent un certain nombre de logiciels pour générer automatiquement des récits. En 1976, Klein et ses collègues mirent au point un logiciel pour simuler les fonctions narratives de Propp et de Lévi-Strauss<sup>23</sup>. Ils y développèrent un langage de règles qui, bien qu'il ne constituât pas une grammaire transformationnelle à proprement

---

Jangfeldt Bengt Rudy Stephen Jakobson, *My Futurist Years*, New York; Saint Paul, MN, Marsilio Publishers, 1997.

20. Noam Chomsky, *Syntactic Structures*, The Hague, Mouton, 1968.

21. Lakoff, « Structural Complexity in Fairy Tales », note 13.

22. Voir le mémorial écrit pour Professor Klein à l'université du Wisconsin et posté à sa mort : <http://www.cs.wisc.edu/news/sklein.memorial.pdf>

23. S. Klein, Aeschliman, Applebaum, Balsisger, Curtis, Foster, Kalish, Kamin, Lee & Price « Simulation d'hypothèses émises par Propp et Lévi-Strauss en utilisant un système de simulation méta-symbolique », *Informatique et Sciences humaines*, n° 28, mars 1976.

parler, en comprenait les éléments essentiels : permutations, substitutions, additions et oblitérations. Ce n'est pas par coïncidence qu'ils citent l'article de Lakoff. Ce système a produit des versions plausibles de soixante contes parmi ceux analysés par Propp et de quatre mythes indigènes analysés par Lévi-Strauss dans *Le Cru et le Cuit*<sup>24</sup>.

L'œuvre de Klein illustre une approche chomskyenne de la **grammaire narrative**. Implicitement, il y a la supposition que la compréhension et la production du langage sont une capacité — un « module », une « faculté linguistique autonome » — qui est spécialisée et distincte des ensembles de compétences nécessaires à d'autres activités que le langage. Lakoff et une cohorte d'anciens étudiants de Chomsky ont rompu avec leur maître sur ce point et ont, finalement, fondé une approche qui, dans les années 1960, s'est appelée la sémantique générative, qu'ont poursuivie ce qu'on a appelé les « linguistiques cognitives ». Les linguistes cognitifs soutiennent que les capacités cognitives utilisées pour traiter le langage sont semblables à celles utilisées pour d'autres tâches non linguistiques<sup>25</sup>. Cela devait accompagner une transformation très radicale de l'approche du récit, qui abonnerait la notion de *transformation* pour celle de *planification*.

LE TEMPS DES PLANIFICATEURS :  
TALE-SPIN ET LA RATIONALITÉ PROCÉDURALE DU RÉCIT

Dès les années 1970, cette importante faction de linguistes était affiliée à un travail continu en intelligence artificielle, ce dérivé de l'informatique fondé en 1956 qui poursuivait un formalisme computationnel pour décrire non seulement la production linguistique, mais aussi toutes les formes d'activité intelligente. Il

24. Si l'on examine leur code (inclus dans l'article publié), la différence entre Klein et Lakoff se trouve dans les détails. Là où Lakoff spéculait sur la possibilité d'écrire un tel logiciel, Klein et ses collègues l'ont fait, et, en l'écrivant, ont identifié un nombre de détails computationnels difficiles négligés dans les analyses de Propp, qui ne reposaient que sur la prose. Voir par exemple *ibid.*, p. 99.

25. Lakoff, par exemple, porte l'attention sur le rôle central du corps humain dans la langue et le discours ; cf. George Lakoff and Mark Johnson, *Metaphors We Live By*, Chicago, University of Chicago Press, 1980.

n'est donc pas surprenant de trouver dans la thèse de James Meehan, écrite en 1976 sous l'auspice du *Yale Artificial Intelligence Project*, un rejet du travail de Klein, au titre que le modèle de ce dernier n'était qu'un modèle linguistique, et non pas un modèle général pour toutes sortes d'actions. Meehan écrit :

« [L'œuvre de Klein] n'est pas un bon modèle ni des histoires ni de l'acte de les raconter ; parce qu'il n'y a pas de causalité impliquée dans les histoires : rien qui nous dise pourquoi une chose en implique une autre, rien qui nous dise ce dont les gens ont besoin, rien qui nous dise pourquoi les gens prennent les décisions qu'ils prennent, rien qui nous explique les réactions des gens les uns envers les autres, rien qui nous représente les complexités du monde matériel. L'objectif de Tale-Spin est justement de prendre en compte ces aspects<sup>26</sup>. »

Tale-Spin : tel était le nom du logiciel de Meehan<sup>27</sup>. Il expose le but de ses recherches ainsi :

« Au cœur de Tale-Spin se trouve un *problem-solver*, un logiciel qui met en œuvre une nouvelle théorie de la planification. En conséquence, les histoires produites par Tale-Spin sont essentiellement des comptes rendus de ce qui est arrivé pendant le processus de résolution d'un ou plusieurs problèmes. Ceci s'accorde avec une théorie qui veut que toutes les histoires ont à voir avec des problèmes<sup>28</sup>. »

De ce point de vue<sup>29</sup>, comprendre l'acte de raconter des histoires est une affaire de planification et de résolution de problèmes, en général, et ne relève pas de quelque faculté spécialisée consa-

26. James Meehan, « The Metanovel : Writing Stories by Computer », Yale University, 1976, p. 9

27. Note du traducteur : « Tale-Spin » est un jeu de mots formé sur les mots anglais *tail* (queue) et *tale* (histoire, conte), avec la formule idiomatique *tailspin*. Celle-ci décrit le mouvement en tire-bouchon d'un avion qui descend en piqué et, en tant que métaphore, exprime une perte de contrôle. « To spin a tale » est aussi une expression idiomatique en anglais pour dire « raconter une histoire ».

28. *Ibid.*, p. 39.

29. Meehan a pris ce point de vue de ses professeurs Roger Schank (un linguiste et informaticien) et Robert Abelson (un psychologue sociale et politologue).

créé exclusivement au traitement du langage<sup>30</sup>. Il s'agit donc d'adapter la forme narrative à ses expressions linguistiques les plus précises. Comme les grammaires transformationnelles de Chomsky, la forme computationnelle de la planification a aussi les moyens de créer des permutations, substitutions, additions et obli-térations. Cependant, contrairement aux grammaires transforma-tionnelles, les plans et la planification étaient considérés comme un formalisme applicable non seulement à la production linguistique mais également à toutes formes de production.

L'histoire du formalisme de la planification peut être trouvée dans les tout premiers articles de Allen Newell et de Herbert Simon présentant le système qu'ils appelaient GPS (General Problem Solver). Le système GPS de Newell et Simon était une formalisation de l'analyse « moyens-fins » (*means-ends analysis*) et fut construit pour jouer aux échecs et trouver la solution à des problèmes mathé-matiques, bien qu'il ait été théorisé comme une mise en œuvre d'un modèle général de la prise de décision que Simon avait développée pour les domaines de l'administration publique, la théorie des orga-nisations, la science politique et l'économie, un modèle dit de la « rationalité limitée » qui permit à Simon de gagner, en 1978, le prix Nobel d'économie<sup>31</sup>. La théorie de la « rationalité limitée » de la

---

30. Dans le chapitre 8 de son livre de 1997, *Computation and Human Experience*, Phil Agre écrit : « Depuis à peu près trente ans, la théorisation computationnelle de l'action a généralement été effectuée sous la rubrique de "la planification". Tandis que d'autres termes computationnels comme "connaissance", "action" et "vérité" nous arrivent chargés d'histoires intellectuelles compliquées, la provenance de "plan" et de "planification" comme termes techniques est facile à tracer. » (Philip Agre, *Computation and Human Experience, Learning in Doing*, Cambridge ; New York : Cambridge University Press, 1997, p. 142). Agre trace cette provenance de 1951 jusqu'à la fin des années 1980 et donc couvre la période dans laquelle travaille James Meehan. Paul Edwards donne un historique détaillé de la planification, qui commence avec le travail de Allen Newell et de Herbert Simon dans son livre, *The Closed World : Computers and the Politics of Discourse in Cold War America*, Inside Technology, Cambridge, Mass., MIT Press, 1996, pp. 230-233.

31. Simon lui-même ne voit aucune différence importante entre son travail sur la résolution des problèmes et la prise des décisions et le pro-gramme de grammaire transformationnelle de Chomsky. Tandis que Chomsky pose une faculté du langage innée, Simon voit la possibilité non seulement d'une faculté du langage, mais aussi de nombreuses autres

prise de décision était sans doute une conséquence de ce qui, dans les sciences sociales, s'est appelé « l'institutionnalisme<sup>32</sup> », une forme d'analyse couramment promue par beaucoup de chercheurs contemporains en sciences sociales connus justement comme les « nouveaux institutionnalistes », ainsi que par certains philosophes contemporains, tel John Searle récemment<sup>33</sup>.

Le système GPS de Newell et Simon utilise une méthode qu'ils appellent « l'analyse moyens-fins<sup>34</sup> ». Simon en explique ainsi le principe :

« Le raisonnement implicite dans GPS est que, si une solution désirée est différente d'une situation présente par la valeur des différences D1, D2, ..., DN, et si l'action A1 enlève les différences de type D1, l'action A2 enlève les différences de type D2, et ainsi de suite, la situation présente peut alors être transformée en la situation désirée si on exécute la séquence d'actions A1, A2, ..., An. [...] Ce raisonnement n'est en aucune manière valable. [...] Sa validité exige des hypothèses fortes quant à l'indépendance des effets de ces diverses actions sur les diverses différences. [...] Or les mondes réels que visent les *problem-solvers* et les "designers" sont rarement complètement additifs en ce sens. [...] Pour cette raison les systèmes de résolution de problèmes et de procédures de "design" dans le monde réel n'assemblent pas simplement les solutions aux problèmes à partir de composants, mais doivent chercher des montages appropriés<sup>35</sup>. »

L'analyse moyens-fins de Simon et de Newell est une conséquence de la « rationalité limitée » pour un certain nombre de rai-

facultés, modules spéciaux de résolution des problèmes applicable à des domaines différents et prêt à s'exécuter dans la cervelle du nouveau-né. Cf., Herbert A. Simon, *The Sciences of the Artificial*, Cambridge, Mass, MIT Press, 1996, pp. 75-77.

32. Voir le discours de Simon pour le prix Nobel, « Rational Decision-Making in Business Organizations », 8 December 1978 à Lindbeck; Assar, *Sveriges riksbank Nobelstiftelsen Economic Sciences, 1969-1980 : The Sveriges Riksbank (Bank of Sweden) Prize in Economic Sciences in Memory of Alfred Nobel*, Singapore; River Edge, N.J., World Scientific, 1992, p. 351.

33. John R. Searle, *Making the Social World : The Structure of Human Civilization*, Oxford; New York, Oxford University Press, 2010.

34. Simon, *The Sciences of the Artificial*, *op. cit.*, p. 121.

35. *Ibid.*, pp. 123-124.

sons. Une des limitations de GPS est son répertoire limité d'actions connues. Ceci fait écho aux observations des **institutionnalistes qui disent que chaque culture a un répertoire limité de façons de faire avancer les choses. Selon l'économiste Douglass North, lauréat du prix Nobel :**

« Les institutions sont les règles du jeu dans une société. [...] Elles sont un guide pour l'interaction humaine, si bien que quand nous désirons saluer des amis dans la rue, conduire une automobile, acheter des oranges, emprunter de l'argent, créer une entreprise, enterrer nos morts, ou quoi que ce soit, nous savons [...] comment effectuer ces tâches<sup>36</sup>. »

La prise de décision par un individu donné sera limitée par le répertoire de moyens, les règles, le répertoire d'institutions auxquels il ou elle a accès dans un contexte culturel donné<sup>37</sup>. De ce point de vue, le reproche que Meehan faisait à Klein et aussi à Propp était qu'ils incluaient dans leurs formalismes trop peu de moyens, d'actions, ou d'institutions. Meehan, à l'inverse, veut inclure des actions et des différences expliquant « pourquoi une chose en implique une autre [...], ce dont les gens ont besoin [et] pourquoi les gens prennent les décisions qu'ils prennent ».

Il y a au moins **trois manières possibles de raconter l'histoire du formalisme**; de l'« analyse moyens-fins », de la « rationalité limitée » et de la « planification » plus généralement. La première commencerait au début de la guerre froide, juste après la fin de la Deuxième Guerre mondiale et décrirait le travail de Newell et de Simon comme un résultat des recherches militaires menées pour

36. Douglass C. North, *Institutions, Institutional Change, and Economic Performance*, Cambridge; New York, Cambridge University Press, 1990, pp. 3-4, p. 6.

37. Les institutions peuvent être formelles (la Constitution des États-Unis) ou informelles (la bonne façon de raconter une histoire à un enfant). Comme l'a démontré le sociologue Ronald Jepperson : « "L'institution" et "l'institutionnalisation" sont des concepts majeurs de la sociologie générale. » Et, « pendant plus d'un siècle, dans le domaine de la sociologie les institutions ont été conçues comme une procédure établie et organisée. Ces procédures spéciales sont souvent représentées comme les règles constituantes de la société (les "règles du jeu"). » Walter W. DiMaggio Paul Powell, *The New Institutionalism in Organizational Analysis*, Chicago, University of Chicago Press, 1991, p. 143.

la RAND Corporation, dont Paul Edwards a fait l'histoire dans son livre *The Closed World*<sup>38</sup>. Telle est aussi la version que raconte Philip Mirowski dans son histoire de l'économie, *Machine Dreams : Economics Becomes a Cyborg Science*<sup>39</sup>. Une deuxième manière reviendrait sur l'analyse des institutions dans la science sociale, et donc commencerait à la fin du XIX<sup>e</sup> siècle avec des figures clés comme l'économiste Vilfredo Pareto, le sociologue Emile Durkheim et l'anthropologue Marcel Mauss. Ce fil historique est suggéré dans le discours de Simon pour le prix Nobel et il transparaît dans les façons dont on utilise le travail de Simon dans la littérature du nouvel institutionnalisme. La troisième manière de raconter cette histoire, encore plus longue, commencerait au siècle des Lumières. Elle est discernable dans la plaisanterie de 1958 de Simon : « [...] les physiciens et les ingénieurs ont eu peu à faire avec l'invention de l'ordinateur numérique — dont le vrai inventeur était l'économiste Adam Smith<sup>40</sup>. » Dans un article publié juste avant sa mort, Simon nous offre une courte esquisse d'une telle histoire qui commencerait avec Smith et Voltaire<sup>41</sup>.

C'est cette troisième histoire explique le mieux l'attrait des formalismes de Simon sur beaucoup de domaines de recherche : « l'analyse moyens-fins » et la « rationalité limitée » sont des réifications de façons de penser anciennes, ayant germé il y a au moins deux cents ans, et non pas de drôles d'oiseaux nés de groupes de réflexion fonctionnant en vase clos, tel que RAND, pendant la guerre froide<sup>42</sup>.

38. Paul N. Edwards, *The closed world : computers and the politics of discourse in Cold War America*, Cambridge, MA, MIT Press, 1996.

39. Philip Mirowski, *Machine Dreams : Economics Becomes a Cyborg Science*, Cambridge; New York, Cambridge University Press, 2002.

40. Herbert, et Newell, Herbert A., « Heuristic Problem Solving : The Next Advance in Operations Research », *Operations Research*, 6, no. January-February (1) (1958).

41. Herbert A. Simon, « Bounded Rationality in Social Science : Today and Tomorrow », *Mind & Society*, 1, no. 1 (2000).

42. Peter Norvig (actuellement Directeur de la recherche à Google), dans son livre *Paradigms of Artificial Intelligence Programming*, retrace l'histoire de l'« analyse moyens-fins » jusqu'à Aristote : « La théorie de l'analyse moyens-fins a été établie avec beaucoup d'élégance par Aristote 2300 ans auparavant dans le livre intitulé « Du courage et de la tempérance » de l'*Ethique à Nicomaque* (Livre III. 5, 1112b.)]. » (Peter Norvig, *Paradigms of Artificial Intelligence Programming : Case Studies in*

## FAUT-IL AVOIR PEUR DES PLANIFICATEURS ?

Il n'est pas nécessaire de croire Simon sur parole pour valider cette généalogie. Le « rêve mécanique » de Simon était en fait le cauchemar de l'Ecole de Francfort. Dans son livre, *The Eclipse of Reason (L'Eclipse de la raison)*, publié en 1946, quelques années seulement avant que Newell et Simon n'inventent leur logiciel d'analyse « moyens-fins », dans le premier chapitre intitulé d'ailleurs « Moyens et fins », Max Horkheimer écrivait :

« Quand on a conçu l'idée de la raison, elle était censée accomplir plus que le simple règlement de la relation entre les moyens et les fins ; elle était considérée comme l'instrument fait pour comprendre les fins, pour les déterminer. Socrate est mort parce qu'il a soumis les idées les plus sacrées et les plus familières de sa communauté à [...], la pensée dialectique [...] Il se battait contre la raison subjective, formaliste préconisé par les autres sophistes<sup>43</sup>. »

*Common Lisp*, San Francisco, CA : Morgan Kaufman Publishers, 1992, p. 111). Il cite le passage suivant d'Aristote : « 11. En outre, nous ne délibérons pas sur les fins à atteindre, mais sur les moyens d'atteindre ces fins. Ni le médecin ne se demande s'il se propose de guérir le malade, ni l'orateur de persuader, ni l'homme politique d'instituer une bonne législation, et ainsi de suite pour le reste où la fin n'est pas en question. Mais, une fois la fin établie, on examine comment et par quels moyens on l'atteindra ; si cette fin paraît devoir être atteinte par plusieurs moyens, on recherche le moyen le plus facile et le meilleur ; s'il n'est qu'un, on recherche comment ce moyen sera atteint, et par celui-là un autre encore, jusqu'à ce qu'on soit parvenu à la cause première, qui est celle qu'on trouve en dernier lieu. Car l'homme qui délibère pousse ses recherches et ses analyses, comme on résout un problème de géométrie. 12. Or, de l'aveu général, toute recherche ne constitue pas une délibération, témoin les mathématiques ; en revanche, toute délibération est une recherche et le dernier résultat de l'analyse se trouve être le premier dans l'ordre de naissance des faits. 13. Si, d'autre part, on se heurte à l'impossible, on renonce au moyen — par exemple, quand on a besoin d'argent et qu'on ne trouve aucune façon de s'en procurer — ; si la chose paraît possible, on se met à l'exécuter. » (Aristote, *Ethique à Nicomaque*, trad. Jean Voilquin, Paris, Garnier-Frères, 1950, pp 101-103.)

43. Max Horkheimer, *Eclipse of Reason*, New York, Oxford University Press, 1947, p. 10

Quelques pages plus loin, il ajoute :

« Plus les idées sont devenues automatiques, instrumentalisées, [et] moins on voit en elles des pensées avec des sens propres à elles. Elles sont considérées comme des choses, des machines. La langue a été réduite à un outil parmi d'autres dans le gigantesque appareil de production de la société moderne<sup>44</sup>. »

Pour Horkheimer, la rationalité formaliste, moyens-fins, automatique, instrumentale, est le côté obscur du siècle des Lumières, ses aperçus et ses méthodes ont été menés trop loin et ont conduit à une situation cauchemardesque.

Sans doute, on ne peut être d'accord à la fois avec Horkheimer et avec Simon. Alors, qu'est-ce qui peut motiver le choix du formalisme de Simon ? Newell et Simon le disaient en 1972 : « [...] une théorie informatique solide de ce qui constitue un bon joueur d'échecs peut jouer une bonne partie d'échecs ; une théorie solide de la création des romans créera des romans<sup>45</sup>. » En d'autres termes, cette théorie semble avoir pour vertu de permettre l'automatisation. Il y a, entre Horkheimer et Simon, un affrontement des valeurs — qui n'est peut-être que la transposition d'un affrontement du monde de l'art et de celui des affaires. Ce qui est vu comme une vertu à partir de la perspective du capitalisme et des affaires pourrait être perçu comme un vice à partir de la perspective de l'art avant-gardiste. Ainsi, le formaliste russe Victor Chklovski déclare que la raison d'être de l'art est de travailler contre l'habitude et l'automatisation :

« L'habitude dévore les œuvres, les habits, les meubles, votre femme, et la peur de la guerre. [...] l'art existe pour que nous puissions récupérer la sensation de la vie ; il existe pour nous faire sentir quelque chose [...], le procédé de l'art est le procédé de singularisation des objets et le procédé qui consiste à obscurcir la forme, à augmenter la difficulté et la durée de la perception. L'acte de perception en art est une fin en soi et doit être prolongé<sup>46</sup>. »

44. *Ibid.*, pp. 21-22.

45. Allen, Simon et Herbert A. Newell, *Human Problem Solving*, Englewood Cliffs, N.J., Prentice-Hall, 1972, p. 11. Cité dans Mirowski, *Machine Dreams : Economics Becomes a Cyborg Science*, pp. 464-465.

46. Viktor Shklovsky, « Art as Technique », dans Lee T. Reis Marion J. Lemon, *Russian Formalist Criticism : Four Essays*, Lincoln : University

ce qui n'est pas compris ici = roman = herméneutique = bon roman vs automatisme = désautomatisation

Or, Vladimir Propp étant un formaliste russe, n'est-ce pas un acte artistiquement pervers d'essayer de coder la *Morphologie du conte* dans un logiciel qui utilise de surcroît le formalisme d'« analyse moyens-fins » de Simon ?

Michel Foucault offre une alternative à cette opposition. Dans son court essai intitulé « Qu'est-ce que les Lumières ? », il propose non seulement une critique indirecte de Horkheimer mais aussi un chemin alternatif. Foucault écrit :

« Mais cela ne veut pas dire qu'il faut être pour ou contre l'*Aufklärung*. ... [Plutôt,] [i]l faut essayer de faire l'analyse de nous-mêmes en tant qu'êtres historiquement déterminés, pour une certaine part, par l'*Aufklärung*. Ce qui implique une série d'enquêtes historiques aussi précises que possible ; et ces enquêtes ne seront pas orientées rétrospectivement vers le “noyau essentiel de rationalité” qu'on peut trouver dans l'*Aufklärung*... ; elles seront orientées vers “les limites actuelles du nécessaire” : c'est-à-dire vers ce qui n'est pas ou plus indispensable pour la constitution de nous-mêmes comme sujets autonomes. [...] il me semble que cette attitude historico-critique doit être aussi une attitude expérimentale. Je veux dire que ce travail fait aux limites de nous-mêmes doit, d'un côté, ouvrir un domaine d'enquêtes historiques et, de l'autre, se mettre à l'épreuve de la réalité et de l'actualité, à la fois pour saisir les points où le changement est possible et souhaitable et pour déterminer la forme précise à donner à ce changement<sup>47</sup>. »

En suivant Foucault, on se propose ici d'expérimenter. On ne doit pas s'abstenir de faire des expériences sur les formes contemporaines de la réalité, même si ces formes sont des conséquences inattendues du projet des Lumières. Ainsi, on peut lire et écrire des logiciels non pas pour améliorer l'emprise technologique sur la réalité, mais « à la fois pour saisir les points où le changement est possible et souhaitable, et pour déterminer la forme précise à donner à ce changement ». Telle est du moins l'interprétation que je propose de ce domaine naissant que l'on appelle les *software*

of Nebraska Press, 1965, pp. 11-12. Voir Victor Chlovski, « L'Art comme procédé », Paris, Allia, 2008.

47. Michel Foucault, « Qu'est-ce que les Lumières ? », in *Dits et Ecrits. Tome II*, Paris, Gallimard, 2001, pp. 1390-91 et p. 1393.

*studies*<sup>48</sup>, qui n'est d'ailleurs qu'une région particulière de ce mouvement en faveur des *digital humanities*<sup>49</sup>.

#### APPRENDRE À VIVRE EN INTELLIGENCE AVEC LES MACHINES

Pour illustrer cela, nous avons mis au point un logiciel très simple que nous avons baptisé *Spinner*, qui est, d'une certaine façon, une mise en œuvre de la *Morphologie* de Propp parce qu'il est, plus précisément, une réinterprétation du logiciel Tale-Spin de James Meehan (1976). On verra qu'une attention scrupuleuse au logiciel ne se justifie pas seulement par l'intention de construire des machines qui permettent de « produire des romans » mécaniquement, comme le voulaient Newell et Simon, mais comme une expérience aux limites de notre modernité. Et, d'une manière remarquable, **réécrire un logiciel est la forme de philologie adéquate aux technologies informatiques.**

En 2008, Noah Wardrip-Fruin écrivait :

« Aujourd'hui, Tale-Spin est une des fictions numériques les plus discutées qui aient jamais été réalisées. Elle n'est pas seulement une pierre de touche en informatique pour les explications de la génération de textes, mais elle est aussi beaucoup citée dans les écrits sur la littérature numérique et le futur de la fiction<sup>50</sup>. »

On pourrait se demander comment, dans le monde si rapidement changeant de la technologie informatique, un logiciel écrit il y a presque quarante ans peut toujours être une pierre de touche. La réponse résulte de questions d'argent.

La recherche sur l'intelligence artificielle (IA) — avant ce qu'on a appelé « l'hiver de l'IA », quand les instances militaires ont drastiquement réduit les budgets alloués à ce champ de recherche — s'est beaucoup intéressée aux récits : il s'agissait

48. Matthew Fuller, *Software Studies : A Lexicon*, Cambridge, Mass., MIT Press, 2008.

49. Susan Schreibman, « A Companion to Digital Humanities », Blackwell; Alliance of Digital Humanities Organizations.

50. Noah Wardrip-Fruin, *Expressive Processing : Digital Fictions, Computer Games, and Software Studies*, Cambridge, MA, MIT Press, 2009, p. 120

d'écrire des logiciels pour comprendre et générer des récits. Mais de la fin des années 1980 jusqu'au milieu des années 1990, certains sujets comme la production informatique de textes ont été largement écartés. Si l'armée avait été jusqu'alors réceptive à l'idée de financer des projets en dialogue étroit avec les sciences humaines — particulièrement en littérature et en philosophie —, les nouveaux sponsors ne voulaient plus que d'une recherche « sérieuse » : par exemple des systèmes experts pour la prospection pétrolière et la détection de fraudes sur les cartes bancaires, des algorithmes de vision artificielle pour la surveillance, etc. Or il se trouve que, récemment, les investissements sont revenus, mais cette fois-ci des entreprises de développement des jeux vidéo : l'art de conter est de nouveau rentable en informatique. C'est la raison pour laquelle ce n'est que maintenant que l'on peut reprendre les recherches pour lesquelles Meehan avait disposé d'un financement<sup>51</sup>.

Cependant, une interprétation de Tale-Spin aujourd'hui ne peut être une reconstruction identique de l'original. D'abord, fondamentalement, Tale-Spin est un planificateur et le dérivé technique de la planification a changé radicalement depuis 1976. Tandis que le formalisme est toujours fortement basé sur l'« analyse moyens-fins » de Newell et Simon, il y a à présent de nouveaux algorithmes pour la mise en œuvre d'un planificateur. Dans

---

51. La recherche dans le domaine de l'*intelligence narrative* est actuellement publiée et partagée dans une variété de lieux et de contextes. Certains sont dédiés précisément à ce sujet, comme les *ACM Intelligent Narrative Technologies Workshops*, l'*International Conference on Interactive Digital Storytelling* (qui a remplacé les deux séries précédentes de congrès européens : *Technologies for Interactive Digital Storytelling* (TIDSE) et *Virtual Storytelling — Using Virtual Reality Technologies for Storytelling* (ICVS)), et le *ACM Multimedia Workshop on Story Representation, Mechanism and Context*. D'autres sont des congrès et des ateliers consacrés plus généralement à l'intelligence artificielle, comme le congrès *Artificial Intelligence and Interactive Digital Entertainment* (AIIDE), ou des ateliers spéciaux comme le *AAAI Workshop on Computational Aesthetics : Artificial Intelligence Approaches to Beauty and Happiness*. Un troisième espace concerne les jeux vidéo : la *Game Developers Conference*, la *ACM Foundations of Digital Games Conference*, le journal *IEEE Transactions on Computational Intelligence and AI in Games*, le *ACM Workshop on Procedural Content Generation in Games*. Cette liste n'est pas exhaustive.

la littérature contemporaine de la planification<sup>52</sup>, les plans ne sont pas habituellement posés en principe comme constructions cognitives, comme l'ont fait Newell et Simon ; on préfère se représenter les systèmes de planification comme une sorte de langages de programmation. Cela nous permet de reconsidérer Tale-Spin non comme une simulation cognitive, mais comme une mise en œuvre d'un estimateur de langage de programmation ou, plus précisément, un planificateur.

Ce que Simon a décrit comme des « actions » dans les systèmes de planification sont représentés en utilisant ce qui est maintenant appelé une notation STRIPS<sup>53</sup>. Les actions, dans cette notation, ont (a) un ensemble de préconditions qui doivent être vraies avant que l'action ne puisse avoir lieu ; (b) un ensemble d'ajouts qui sont des termes ajoutés à la base de données après que l'action a été réalisée ; et (c) un ensemble de retraites qui sont des termes retirés de la base de données après que l'action a été réalisée. En plus des actions, le langage de planification implémenté prend également en charge les méthodes, qui sont définies comme séquences d'actions.

Le langage de planification que nous avons utilisé pour réécrire Tale-Spin est un planificateur HTN, Hierarchical Task Network (Réseau hiérarchique de tâches). La planification HTN fut développée pour d'autres usages à l'université du Maryland<sup>54</sup>, mais elle s'est révélée très utile pour l'animation de personnages autonomes dans les jeux vidéo<sup>55</sup> et pour la génération de discours de personnages<sup>56</sup>. Le planificateur HTN sera ici implémenté sous

52. E.g., Ghallab, Malik ; Nau, Dana S. ; Traverso, Paolo, *Automated Planning Theory and Practice* (Elsevier/Morgan Kaufmann, 2004).

53. Fikes, R. et Nilsson, N., « Strips : A New Approach to the Application of Theorem Proving to Problem Solving », *Artificial Intelligence*, 2 (1971).

54. D. Nau, « Shop2 : An Htn Planning System », *Journal of Artificial Intelligence Research* 20 (2003).

55. J. P. ; Botea Kelly, A. ; et Koenig, S., « Planning with Hierarchical Task Networks in Video Games », in *ICAPS-07 Workshop on Planning in Games* (2007).

56. Strong, Christina R. et Mateas, Michael, « Talking with Npcs : Towards Dynamic Generation of Discourse Structures », dans *4th Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE 2008)* Palo Alto, CA2008.

forme d'une recherche à travers un espace d'états où chaque état est, lui-même, représenté comme une base de données d'assertions sur la diégèse (le monde du récit). Chaque fois qu'une action est exécutée, les ajouts et les retraites de l'action changent la base de données et donc forment un nouvel état qui suit l'état précédent. On peut comprendre cet espace ramifié d'états du monde comme des ensembles de « mondes possibles » au sens où l'on en parle en philosophie et en littérature<sup>57</sup>. Le planificateur fait une recherche dans l'espace des états en appliquant des combinaisons de méthodes et d'actions jusqu'à ce que toutes les tâches aient été réalisées, soit n'ait pu nul état final être atteint. Cette façon de refondre un logiciel de résolution de problèmes pour en faire une recherche dans un espace abstrait d'états reste une technique bien connue dans l'IA que Newell et Simon ont décrit pour la première fois il y a maintenant des décennies<sup>58</sup>.

Actuellement, la *lingua franca* des structures de données échangées entre les sites Web et les langages de programmation est JSON, la JavaScript Object Notation<sup>59</sup>. JSON a été formalisé et popularisé par Douglas Crockford, qui travaille désormais chez PayPal. JSON définit une syntaxe qui est surtout utilisée comme format d'échange de données (entre les clients HTTP et les serveurs et entre les serveurs Web). Cette syntaxe est employée dans cette interprétation de Tale-Spin. La base de données utilisée dans le système est une base de données NoSQL « orientée documents » qui stocke et recherche des objets JSON.

Envisageons d'abord une tâche plus simple que la génération d'une histoire pour bien comprendre comment fonctionne un planificateur HTN. Entasser des blocs sur une table est un jeu canonique qui est utilisé depuis des décennies dans la littérature de la planification pour illustrer les capacités d'un planificateur. L'état initial est enregistré dans la base de données avec un ensemble d'assertions indiquant quels blocs sont sur la table et lesquels sont sur d'autres blocs. L'état visé, symétriquement, est une liste d'assertions déclarant quels blocs devraient être à quels endroits. Pour cette tâche, il

---

57. Cf., Marie-Laure Ryan, *Possible Worlds, Artificial Intelligence, and Narrative Theory*, Bloomington : Indiana University Press, 1991.

58. N. J. Nilsson, *Problem-Solving Methods in Artificial Intelligence*, New York : McGraw-Hill, 1971.

59. Douglas Crockford, <http://www.json.org/>

n'y a essentiellement qu'une action qui doit être définie ; une action qui s'appelle « *move* » (déplacer). Avant qu'un bloc puisse être bougé, il doit être « *clear* » (dégagé). Un bloc est « *clear* » s'il n'y pas d'autres blocs dessus. Une règle consultant la base de données décide si un bloc est dégagé. Après qu'un bloc est déplacé, une assertion est supprimée de la base de données et une autre y est ajoutée. L'assertion supprimée indique sur quoi le bloc était auparavant, la nouvelle sur ce quoi le bloc est actuellement posé. Cette implémentation contient aussi un ensemble de méthodes, c'est-à-dire de séquences d'actions. Par exemple, la première méthode employée par le planificateur est « *unstackAll* » (désentasseTout) : d'abord, tous les blocs sont mis sur la table pour qu'ils soient tous libres d'être déplacés ailleurs.

Voilà une illustration du planificateur. L'état initial est défini par cet ensemble d'assertions :

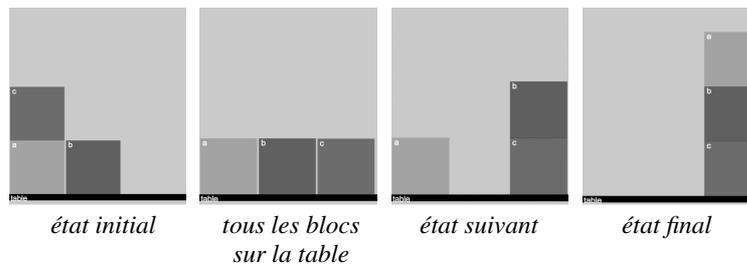
```
{ « on » : { « above » : « b », « below » : « table » } },
{ « on » : { « above » : « a », « below » : « table » } },
{ « on » : { « above » : « c », « below » : « a » } }
```

L'état final désiré est défini avec une série de buts (« *goals* ») :

```
{ « goal » : { « on » : { « above » : « a », « below » : « b » } } },
{ « goal » : { « on » : { « above » : « b », « below » : « c » } } },
{ « goal » : { « on » : { « above » : « c », « below » : « table » } } }
```

La méthode « *unstackAll* » est utilisée par le planificateur pour déplacer tous les blocs jusqu'à la table. Alors, le planificateur utilise l'action « *move* » de multiples fois pour atteindre l'état final désiré<sup>60</sup>.

60. Une explication plus détaillée et le code source complet pour cet exemple peuvent être trouvés en ligne ici : Warren Sack, « HTN Planner » <http://softwarearts.info/Code/Plan/> Une démonstration animée du planificateur en train d'effectuer cette tâche peut être trouvée ici : « HTN Planner Demonstration », <http://softwarearts.info/Code/Plan/Code/Plan/planAndDisplayBlocks.html>



Maintenant, voyons ce que cela peut donner pour la génération automatique de récits. Les personnages et leurs attributs sont introduits par des assertions au sein de la base de données. Des règles de production définissent différentes sortes de personnages. Quand un personnage est défini comme étant d'une sorte particulière, par exemple tel genre d'animal, une règle de production ajoute un ensemble d'attributs au personnage. Par exemple, si un personnage a dit être un ours, une règle de production pourrait ajouter des assertions telles que « le personnage a de la fourrure, des griffes et des crocs ». Un autre ensemble de règles de déduction explicite le fonctionnement du monde de l'histoire. Par exemple, une règle de déduction pourrait déclarer que si deux personnages sont au même endroit, ils connaissent chacun la position de l'autre. Ou bien, une autre règle de déduction pourrait déclarer que si un personnage menace de tuer un autre ou de lui voler quelque chose, cette menace est une menace sérieuse (ne doit pas être prise à la légère). Un ensemble d'actions de planification est défini pour donner, prendre, marcher, informer, voler dans le ciel, boire, manger, etc. Si l'on suit la discussion par John Searle de la différence entre ces deux actes de langage que sont la menace et la promesse, on définira les deux de la même manière, à ceci près que, dans une menace, le locuteur s'engage à faire quelque chose *au* destinataire, alors que, dans une promesse, le locuteur s'engage à faire quelque chose pour le destinataire. Un ensemble de méthodes de planification définit les séquences d'actions. Par exemple, la méthode du troc entraîne une série d'actes de langage entre deux personnages et en fin de compte un échange d'argent ou de marchandises.

On donne au planificateur une série de tâches — ou, dans le langage de Propp, une série de fonctions — et l'algorithme de génération de textes fait une recherche pour une série d'actions qui puisse réaliser ces tâches. Par exemple, si on donne au planifica-

teur deux tâches — (i) Maître Reynard, le renard, a besoin d'obtenir du fromage et (ii) de manger le fromage —, le logiciel utilise un ensemble d'actions et de méthodes (y compris une méthode pour duper) pour générer cette version déformée d'une des fables d'Esopé : « Le Renard et le Corbeau ».

« Un jour... maître reynard voulait avoir le fromage. maître reynard voulait aller au corbeau. maître reynard s'est déplacé de l'orme jusqu'au chêne. maître reynard a marché de l'orme jusqu'au chêne. maître reynard était à côté du chêne. maître reynard croyait que maître reynard était à côté du chêne. maître reynard a demandé à corbeau pour courbeau de chanter quelque chose à quelque chose. corbeau croyait que maître reynard désirait que corbeau chante quelque chose à quelque chose. corbeau a chanté quelque chose à quelque chose. corbeau a laissé tomber le fromage. fromage était à côté du chêne. corbeau a informé maître reynard que quelque chose. maître reynard croyait que quelque chose. maître reynard a volé le fromage de corbeau. maître reynard a porté le fromage. maître reynard a mangé le fromage<sup>61</sup>. »

\*

Nous n'avons évidemment présenté et discuté ici que quelques-unes des centaines de lignes de code nécessaires pour implémenter Spinner, l'interprétation de Tale-Spin. Nous avons négligé les détails du mécanisme qui produit le texte que vous venez de lire, mais il n'est pas différent, dans ses principes, de ce que nous avons discuté à propos de l'« analyse moyens-fins ».

Le théoricien des nouveaux médias Noah Wardrip-Fruin a fait remarquer que les logiciels de ce genre, et Tale-Spin en particulier, font l'objet de réception nettement contrastée dans la communauté informatique et au sein des humanités :

61. Le code qui implémente cette réécriture de Tale-Spin peut être trouvé ici : « Spinner Source Code » *view-source:softwarearts.info/Code/Narrative/Spinner/spinner.html*

Pour exécuter ce code et produire une histoire, ouvrez d'abord la console JavaScript, et puis consultez cette page : « Spinner Demonstration », *http://softwarearts.info/Code/Narrative/Spinner/spinner.html*. La console JavaScript doit être ouverte dans la fenêtre pour que l'on voie les données en sortie du générateur d'histoires.

« La discussion continue de [Tale-Spin]... a été rendue possible par deux faits. D'une part, circulent de nombreux exemples de fictions produites par Tale-Spin. Ce sont eux qui motivent les discussions de Tale-Spin dans le monde des humanités, et elles ont tendance à être méprisantes. D'autre part, la thèse de Meehan rend compte de manière détaillée des opérations des processus de Tale-Spin, ainsi que des informations importantes à propos de ses données. Cela sert de base à la plupart des discussions de Tale-Spin en informatique, qui ont tendance à traiter le système comme digne d'une attention sérieuse<sup>62</sup>. »

En d'autres termes, les uns et les autres ne parlent pas de la même chose. On peut en effet juger un tel logiciel soit à ses résultats, soit à sa technique. Les « exemples de fictions produites par Tale-Spin » sont du genre du résultat de Spinner présenté ci-dessus, en l'occurrence la version déformée de la fable du « Renard et le corbeau ». Ils paraissent évidemment ennuyeux et sans intérêt tant qu'ils sont jugés comme des morceaux de prose. Mais Wardrip-Fruin semble suivre les formalistes russes en nous encourageant à ne pas regarder que la prose et à prendre en compte aussi le procédé, en l'occurrence le logiciel, le code, le mécanisme :

« Essayer d'interpréter une œuvre des médias numériques en regardant seulement les *outputs* revient à interpréter un modèle du système solaire en ne regardant que les planètes. S'il s'agit de vérifier l'exactitude du rendu de la texture de la surface de Mars, cela peut aller. Mais ça ne suffira pas si nous voulons savoir si le modèle illustre et met en œuvre une théorie copernicienne — ou à l'inverse place la terre au centre de son système solaire simulé. Les deux sortes de théories pourraient produire des modèles qui placent actuellement les planètes dans des emplacements appropriés, mais l'examen des fils et des mécanismes des modèles révélera des différences cruciales, et probablement les différences qui en diront le plus long<sup>63</sup>. »

C'est ce que nous avons remarqué à travers notre petite histoire : des formalismes bien différents ont pris en charge le

62. Wardrip-Fruin, Noah, *Expressive Processing : Digital Fictions, Computer Games, and Software Studies*, Cambridge, MA, MIT Press, 2009, p. 121.

63. *Ibid.*, p. 158.

on a ici affaire à un malentendu entre compréhension et surpréhension.

projet de mettre en place une machine à raconter des histoires, et ces différences sont profondément significatives : considère-t-on un récit comme une variante sur un seul schéma (Propp) ou comme la transformation d'un autre récit (Lévi-Strauss) ? Faut-il y voir un système de réécriture qui passe d'un état à un autre par des règles de « transformation », ou bien une forme de résolution d'un problème ? Seule une attention précise aux détails du fonctionnement permettra d'aborder ces questions.

Il est nécessaire de distinguer la conception d'un logiciel de son utilisation. Du point de vue de l'utilisateur, un logiciel n'est qu'une forme d'automatisation, une façon de faire rentrer dans une routine, c'est-à-dire d'enfermer, dans une « boîte noire », une activité habituelle ou une institution conventionnelle. Mais, le travail que fait un programmeur pour réécrire un processus dans la forme d'un logiciel demande qu'il repense et réarticule cette même activité habituelle ou institution conventionnelle. Cela explique que les seconds trouvent matière à émotion là où les premiers ne voient que banalités et monstruosité. Comme le disait Chklovski :

« La littérature est sans pitié ou étrangère à la pitié, hormis les cas où le sentiment de compassion est employé à une construction. Mais même là, quand on en parle, il faut l'aborder du point de vue de la composition, tout comme il faut, si vous voulez comprendre une machine, regarder la courroie de transmission comme une pièce de la machine, et non pas la considérer d'un point de vue de végétarien<sup>64</sup>. »

Et on aurait tort de se contenter de cette dualité de points de vue. Pour l'utilisateur, les logiciels sont censés n'être qu'un automatisme qui, selon les doctrines contemporaines du design des interfaces, ne devrait pas nous pousser à trop réfléchir ; ils devraient, selon ces doctrines, être « intuitifs » et « faciles à utiliser ». En revanche, pour un programmeur, la composition de logiciels offre un défi à la forme habituelle de la pensée. Le processus de la composition de logiciels est l'opposé de ce que les concepteurs demandent et ce que les « utilisateurs » attendent des logiciels. Au fond, le paradoxe des technologies de simulation est qu'elles doivent masquer la logique de leur fonctionnement pour

64. V. Chklovski, *Sur la théorie de la prose*, Lausanne, L'Age d'Homme, 1973, p. 230.

faire croire qu'elles épousent celle des utilisateurs, alors qu'en réalité elles lui imposent une logique procédurale bien plus radicale. On aurait donc tort d'attendre simplement des machines à raconter les histoires, qui commencent à envahir notre environnement, qu'elles finissent par effacer les bizarreries comme celles de la fable produite par Spinner. Au contraire, on doit s'appuyer sur des versions particulièrement frustrées et contre-intuitives du récit pour prendre conscience de la véritable logique qui les anime. Car une conclusion s'impose : l'effort pour traduire le vieil art narratif dans la technologie informatique ne se contente pas de mécaniser la compétence narrative ; elle transforme la nature des récits racontés. Ce que notre petite histoire de l'« intelligence narrative » aura voulu montrer, ce sont à la fois la nécessité et les difficultés qu'a rencontrées l'idée même d'une automatisation de l'art narratif. Il se trouve que celle-ci a trouvé dans la forme de la « planification » une forme relativement efficace. Mais cette traduction ne laisse pas intactes les logiques narratives : simuler, ici, c'est imposer. Notre version de la fable du Renard et du Corbeau n'est sans doute pas très charmante, mais elle a le mérite de *montrer* la logique à l'œuvre dans la production des récits à l'âge numérique. Si l'on veut regagner de la liberté, des « marges d'actions » — comme eût dit Foucault — dans l'exercice de l'art narratif, qui dans l'avenir passera très largement par les technologies numériques, il faudra apprendre à améliorer le logiciel. Aussi est-il nécessairement de comprendre qu'il n'y a pas de fatalité dans l'implémentation informatique d'une compétence naturelle, mais un ensemble de choix techniques réversibles, à condition qu'on trouve l'intelligence technique de parvenir à des solutions de remplacement. La planification est la contingence de la narration automatique dans son état actuel. Doit-on répondre comme dans le poème de Char que Foucault avait mis en quatrième de couverture de son *Histoire de la sexualité* : « Y contredire est un devoir » ? Nous laisserons le lecteur juger par lui-même.

Ainsi, au lieu d'opposer les analyses « formalistes » et « culturalistes », il est temps de reconnaître que le point de vue du concepteur de logiciels — comme celui du mécanicien de Chklovski — comprend une forme de pensée. Les *software studies* ne sont pas la même chose que, par exemple, les *cultural studies* ; elles ne sauraient apporter ce qu'une méthode d'inspiration bakhtinienne peut apporter dans l'approche de la littérature. Mais

elles ne sont pas non plus en opposition directe. Elles proposent une approche complémentaire — une approche qui nous invite à interroger la technologie familière qui nous entoure, à ne pas la tenir pour acquise, à ouvrir les boîtes noires —, afin de se donner les moyens, si on le souhaite, de faire face aux périls éthiques qui découlent toujours de l'interdit de penser<sup>65</sup>.

Warren SACK

*Traduit de l'anglais par Sophie Bargues Rollins*

---

65. Roger Katz Jeffrey Keenan Thomas Berkowitz, *Thinking in Dark Times : Hannah Arendt on Ethics and Politics*, New York, Fordham University Press, 2010.