

Apprentissage du russe par ordinateur

Pierre Baccheretti

Citer ce document / Cite this document :

Baccheretti Pierre. Apprentissage du russe par ordinateur. In: Revue Russe n°1, 1991. La société Soviétique d'Aujourd'hui. Actes du colloque d'Aix-en-Provence, 1er et 2 décembre 1990. pp. 123-137;

doi : <https://doi.org/10.3406/russe.1991.1749>

https://www.persee.fr/doc/russe_1161-0557_1991_num_1_1_1749

Fichier pdf généré le 30/03/2018

Pierre BACCHERETTI
(Université de Provence)

**APPRENTISSAGE DU RUSSE
PAR ORDINATEUR**

Au cours des dernières années les logiciels informatiques éducatifs, dits aussi didacticiels, sont de plus en plus utilisés dans le domaine de l'apprentissage des langues. Le russe ne fait pas exception à la règle, et, même si à l'heure actuelle la plupart des éditeurs en informatique continuent à considérer qu'il s'agit d'un "créneau" sans le moindre intérêt commercial, on sait que de nombreux collègues se servent de l'informatique de façon régulière dans leur enseignement.

Toutefois, il n'existe actuellement, à notre connaissance, aucun logiciel consacré spécialement à l'apprentissage de l'ensemble de la déclinaison en russe. Alors que l'expérience de tous les jours montre à l'évidence qu'il s'agit d'un domaine dans lequel nos élèves et étudiants ont besoin d'une aide permanente et manquent d'outils de travail facilement et immédiatement accessibles.

De ce constat est née l'idée de tenter de réaliser un logiciel spécialisé dans l'apprentissage de la déclinaison, susceptible de pouvoir venir en aide à n'importe quel utilisateur, indépendamment de son niveau de connaissances et fonctionnant sur une machine d'un prix raisonnable accessible à l'élève ou l'étudiant moyen. Sans même parler de sa conception technique, la réalisation d'un tel logiciel posait de nombreux problèmes, entre autres, d'ordre pédagogique: quel devait être "le cahier des charges" d'un tel logiciel, à quels utilisateurs devait-il s'adresser, que devait-il pouvoir leur offrir pour répondre à leur attente, quels devaient être respectivement les rôles et les attributions de l'utilisateur et de la machine?

La démarche pédagogique: une approche différente

les logiciels traditionnels.

La première question que nous nous sommes posée a été celle du corpus que devait pouvoir traiter notre futur programme. La réponse

était fonction de la réponse fournie à une autre interrogation: à quel public devait-il s'adresser?

Il se trouve, en effet, que la plupart, sinon la totalité, des programmes existant en langue visent un public très ciblé, défini par un niveau de compétence qui s'exprime en termes de parcours scolaire ou universitaire. Les titres sont, à cet égard, très révélateurs: "*langue x au collège, langue y au lycée, langue z au baccalauréat*, etc. Et le principe de fonctionnement ne varie guère: en schématisant, nous dirons qu'un ensemble de données, ordonnées selon un modèle type question/réponse a été introduit dans le programme par son concepteur, la machine se contentant de proposer à l'utilisateur de façon plus ou moins aléatoire la question x du programme dans l'attente de la réponse correspondante y, au besoin, en analysant et corrigeant une éventuelle réponse erronée.

Les avantages d'une telle approche sont évidents: la gamme des questions et réponses est répertoriée à l'avance et ne laisse la place à aucune possibilité d'erreur lors de l'impression finale à l'écran. Toute une progression est possible dans l'ordre de présentation des questions et des réponses. Tel thème peut être approfondi de préférence à tel autre, etc. Les inconvénients ne sont pas moindres: à aucun moment l'utilisateur ne peut "mener le jeu", prendre l'initiative de poser telle ou telle question de son choix qui n'aurait pas été prévue de façon explicite par le concepteur du programme. Ainsi, en termes de lexique, le programme se limitera-t-il à un corpus fermé (d'importance variable selon le programme et la machine), par exemple, le lexique qu'il est prévu d'utiliser dans telle ou telle classe selon les instructions de l'inspection Générale des Langues Vivantes, ou encore le lexique de tel ou tel manuel scolaire, de tel ou tel domaine des connaissances, etc.

Une telle approche suppose donc la confection d'un nombre x de programmes parallèles, à l'usage exclusif de catégories limitées d'utilisateurs qui, en fin de compte, n'ont guère le droit à la parole et veulent bien accepter passivement ce qui leur est proposé de façon arbitraire.

Un programme "ouvert":

Quel public?

Nous avons souhaité pour notre part réaliser un programme "ouvert" qui ne tienne pas compte du niveau de compétence linguistique de l'utilisateur, autrement dit, un programme évolutif qui puisse s'adresser aussi bien à l'élève débutant du collège qu'à l'étudiant du 1-er ou 2-ème cycle universitaires.

Nous avons surtout souhaité que ce ne soit plus la machine qui questionne en fonction de réponses préétablies, mais bien l'utilisateur en fonction de ses désirs et intérêts du moment. Il fallait donc que le programme puisse traiter n'importe quel terme proposé, c'est-à-dire en indiquant les diverses formes casuelles, en fonction d'un modèle informatique de référence. Dans ces conditions le seul corpus acceptable devenait l'ensemble des éléments de la langue russe soumis à déclinaison, autrement dit, l'ensemble des substantifs, pronoms, adjectifs et formes assimilées. Dans le cas de l'adjectif, il était évident qu'il ne devait pas être traité isolément, mais comme faisant partie d'une séquence adjectif+substantif accordée, et c'était au programme de prendre en charge cet accord à partir des constituants isolés fournis par l'utilisateur.

Les rôles respectifs de l'utilisateur et de la machine étaient donc définis: l'utilisateur proposait les éléments à traiter à sa seule convenance, la machine après analyse et traitement lui fournissait l'ensemble de la déclinaison du substantif ou de la séquence adjectif/substantif ou pronom/substantif considérés.

Quel contenu?

Qu'entendons-nous par " l'ensemble de la déclinaison"? Dans un premier temps, bien évidemment, la totalité des désinences casuelles de singulier et de pluriel propres à la catégorie linguistique à laquelle appartient le terme proposé par l'utilisateur, mais également les réalisations variables de ces désinences catégorielles dans le cas du terme concerné, qu'il s'agisse de variations prévisibles du fait de la nature de la consonne ou voyelle finale du radical, ou d'irrégularités consacrées par l'usage. Mais aussi les réalisations concurrentes en fonction de l'environnement dans la phrase (ainsi pour certains substantifs un traitement différent après les prépositions gouvernant le locatif ou le prépositionnel). Mais aussi les modifications du radical - la "voyelle mobile", par exemple - qui interviennent ou n'interviennent pas avec certains suffixes, ou dans certaines positions lorsque la désinence est la désinence zéro. Le programme doit également pouvoir indiquer à l'utilisateur que le terme qu'il a proposé ne rentre pas dans un paradigme russe et reste indéclinable. Ou encore qu'il ne possède que les formes du pluriel, ou n'est utilisé pratiquement qu'au singulier. Ou enfin qu'il présente des formes concurrentes.

La correction automatique, ses limites:

Une telle conception du programme entraînait un certain nombre de contraintes, souvent contradictoires. Du fait de la liberté totale de choix

laissée à l'utilisateur, un problème se présentait immédiatement qu'il était difficile de résoudre. Si le ou les modèles informatiques auxquels se réfère le programme sont suffisamment fiables, aucune erreur n'interviendra dans le traitement du mot proposé. Mais comment le programme pourra-t-il déceler la présence d'erreurs dans le terme saisi au clavier par l'utilisateur et, au besoin, les lui signaler, ou mieux encore, les corriger de lui-même, sans intervention extérieure.

La réponse n'est pas univoque. La machine, le programme pourront identifier des séquences de signes qui ne peuvent se réaliser dans le système de la langue, ainsi une graphie *ja* après une chuintante ou une vélaire. Mais, à moins qu'on ne lui ait préalablement introduit en mémoire un dictionnaire orthographique complet de la langue russe, l'ordinateur sera dans l'impossibilité de déceler un mot fictif dont la graphie serait conforme aux normes. Cette solution, tout à fait envisageable sur un gros ordinateur disposant d'une capacité de mémoire importante, ne paraissait guère réalisable dans le cas d'un logiciel individuel à l'usage d'élèves ou d'étudiants.

Il nous a paru, toutefois, indispensable de donner à la machine les moyens de déceler, sinon un terme fictif, du moins tout terme dont la désinence répertoriale serait orthographiée de façon incorrecte. Dans ces cas de figure, le programme identifiera la graphie incorrecte de la désinence, signalera la chose à l'utilisateur, et lui fournira la graphie correcte, avant de procéder au traitement du terme. Ce type de correction automatique reste, par définition, limité aux seuls adjectifs (ou formes assimilées, tels les participes) et ne concerne que les seules désinences. Les erreurs qui interviendraient dans le corps du terme proposé, pour peu qu'il n'y ait pas d'incompatibilités orthographiques, ne seront pas reconnues. Ainsi une forme *horošyj* sera identifiée comme incorrecte et corrigée par la machine, qui acceptera néanmoins et traitera une forme *harošij*.

Pour éviter toute ambiguïté sur ce point, un message intervient systématiquement, dès la saisie au clavier, pour signaler à l'utilisateur que le ou les termes qu'il a proposés ne sont pas contrôlés par la machine et qu'il peut s'agir de formes fictives, proposées sous sa seule responsabilité.

Il s'agit là d'un inconvénient certain, mais qui découle de la conception même du programme, dont le but n'est pas de procéder à un apprentissage du lexique, mais d'entraîner l'utilisateur à pratiquer (par le biais de la déclinaison) un lexique déjà connu de lui.

Cet inconvénient peut, d'ailleurs, devenir un avantage, car il devient possible d'étudier les diverses réalisations d'une même désinence en imaginant un radical fictif, par exemple, la séquence de signes *****, puis en lui adjoignant une consonne (****-k*) ou une voyelle finale (****-a*) que l'on fera varier à volonté, pour, ensuite, dégager les règles qui

président aux diverses réalisations de la désinence en fonction de l'environnement linguistique. Il est bien évident que cette approche n'est accessible qu'à un utilisateur déjà averti, ou nécessite l'aide d'un enseignant. Le programme permet, alors, de mettre en évidence les structures profondes de fonctionnement de la langue en faisant abstraction des réalisations ponctuelles, et, à ce titre, permet de développer chez l'utilisateur une réflexion de caractère généralisateur et théorique qui donnera une assise solide à la création d'automatismes.

Utilisation passive et active

Tout au long de la réalisation du programme, nous avons voulu que l'utilisateur puisse, à chaque instant, choisir son degré de participation.

Il peut s'adresser au programme à titre purement consultatif pour connaître immédiatement la déclinaison d'un mot ou d'une séquence de mots qu'il veut utiliser dans un exercice grammatical, dans un thème, etc. Son rôle est alors purement passif, et le logiciel peut être assimilé à une sorte de dictionnaire des formes grammaticales directement et immédiatement consultable.

L'utilisateur peut, au contraire, vouloir travailler de façon systématique telle ou telle déclinaison saisie de façon globale. Il propose un mot, l'ordinateur lui demande de fournir successivement les diverses formes casuelles, lui signale ses erreurs éventuelles, et au bout de 3 essais infructueux affiche à l'écran la forme correcte. L'utilisateur intervient dès lors de façon active, peut tester ses connaissances, perfectionner ses automatismes.

L'interrogation peut se faire en faisant apparaître les divers cas dans l'ordre traditionnel des grammaires ou, à l'inverse, pour éviter que ne joue un phénomène de mémoire répétitive, dans un ordre aléatoire.

En fin de parcours, la machine affiche le score réalisé (le nombre de bonnes réponses fournies) et attribue une note sur 20. Les expériences menées avec un public d'âge scolaire montrent que cet aspect ludique du programme génère de sa part une attention particulièrement soutenue (il s'agit d'un défi et il faut arriver à battre la machine!) et débouche sur une mémorisation d'autant plus rapide que l'élève n'a plus le sentiment de pratiquer un exercice scolaire et fixe ces nouvelles acquisitions de façon inconsciente, donc plus en profondeur.

La solution qui, à l'expérience, nous apparaît comme la plus efficace consiste à utiliser successivement les deux options de travail: dans un premier temps en consultant passivement l'ordinateur pour avoir un modèle, puis en reprenant immédiatement et de façon active le même terme. En quelque sorte, le cours et les travaux pratiques!

Une autre approche est prévue qui fait l'objet d'un module particulier et dans laquelle le mode de travail n'est plus la déclinaison complète d'un seul terme, mais l'étude d'un cas déterminé, ce qui permet de faire ressortir les différences spécifiques de désinences pour un cas donné entre des termes appartenant à des catégories différentes.

Un environnement russe

Nous avons choisi de poser les questions, non pas à partir du nom des cas qui risque de rester une notion trop abstraite pour beaucoup d'utilisateurs débutants, mais à partir des interrogatifs accordés, de façon à reproduire, autant que faire se peut, les conditions d'un dialogue réel. Ainsi le nominatif sera-t-il identifié par la question *kogo / čto*, l'accusatif par *kogo / čto*, le datif par *komu / čemu*, etc.

Précisons que l'ensemble des interrogations qui interviendront dans la zone de dialogue en bas d'écran seront rédigées en russe. Le fait de travailler en permanence dans un environnement en langue étrangère nous paraît, en effet, un gage certain d'efficacité. Les débutants éprouveront, sans doute, quelques difficultés au début, mais les mêmes questions revenant en permanence, cette difficulté disparaîtra au bout de quelques heures de pratique, d'autant plus qu'un récapitulatif des questions possibles, rangées par ordre alphabétique, est donné dans le manuel d'accompagnement du logiciel.

Un programme interactif: avantages et inconvénients

Il est bien évident que par sa conception même le programme n'est pas à même de traiter seul l'ensemble des termes qui lui seront soumis de façon totalement aléatoire par l'utilisateur. La machine ne va pas travailler sur des mots réels répertoriés en mémoire et donc identifiables, mais sur des séquences de signes, saisis au clavier, puis transformés en chiffres et traités mathématiquement. Or la graphie d'un terme ne peut à elle seule rendre compte de toutes les caractéristiques indispensables à son fonctionnement dans la langue.

Ainsi la valeur sémantique du terme et, partant, son appartenance à la sous-catégorie animé/inanimé, ne peuvent être déduites de sa graphie. Il en est de même, dans quelques cas isolés, pour les diverses réalisations graphiques de la désinence casuelle en fonction de la place de l'accent tonique. Le genre du mot, enfin, ne peut se déduire de sa forme graphique que dans un nombre de cas limité. Comment la machine, à moins qu'on n'ait introduit chacun de ces mots dans sa mémoire, pourrait-elle faire la différence entre, par exemple, "stol" inanimé et "vol" animé, entre "otec" à accent désinentiel et "ranec" à accent

radical, entre "den" masculin et "ten" féminin? Et lorsqu'elle devra traiter une séquence adjectif+substantif et en accorder les éléments, comment différenciera-t-elle ces mêmes derniers mots ou des termes comme "papa" et "mama"?

Elle va donc être amenée à poser à l'utilisateur une ou plusieurs questions qui lui permettront de rattacher le terme proposé à tel ou tel modèle de référence et de le traiter en conséquence.

Les questions porteront essentiellement sur deux points, incontournables par le programme, tel qu'il est conçu. D'une part le genre des substantifs terminés en , d'autre part l'accent, lorsqu'il a un rôle à jouer dans la réalisation graphique du phonème /o/, qu'il s'agisse du substantif ou de l'adjectif.

Cette démarche pourrait apparaître comme un défaut du programme dans la mesure où, en cas de réponse erronée de l'utilisateur, le terme proposé sera rattaché à un modèle existant, mais qui ne le concerne pas, et la forme affichée à l'écran par la machine ne correspondra pas à la réalité de la langue.

Toutefois cette difficulté existe aussi dans la réalité de la pratique en langue étrangère (comment utiliser "*rol*" si on ne sait pas qu'il s'agit d'un féminin!), et on peut penser que la répétition systématique de la même question à propos d'une catégorie de termes donnée ne peut que sensibiliser l'utilisateur aux difficultés propres à cette catégorie, l'amener à formuler plus clairement dans son esprit les règles qui jouent dans le cas de figure considéré.

Nous pensons, par ailleurs, que fournir dans tous les cas des réponses toutes prêtes à l'utilisateur, en éliminant ou en masquant les difficultés, n'est pas la meilleure démarche pédagogique, et qu'il vaut mieux l'amener à prendre conscience de la réalité de ces difficultés et lui apprendre à les résoudre de façon consciente et réfléchie plutôt que de ne faire appel qu'à sa seule mémoire.

Il est, d'ailleurs, à noter que dans une adaptation du programme qui fonctionnerait sur un gros ordinateur disposant d'une mémoire plus étendue, il serait facile de pallier cet inconvénient supposé en recourant à des fichiers exhaustifs, ou à une analyse plus poussée des caractéristiques inhérentes à tel ou tel suffixe.

Nous avons, cependant, imaginé une amorce de solution immédiate au problème. En même temps que le programme de déclinaison pourront être fournis un ou plusieurs programmes annexes, en fait des fichiers de données, dans lesquelles l'utilisateur ira chercher, s'il y tient absolument, des séries de termes répertoriés, correspondant au programme officiel de telle ou telle classe du secondaire, au lexique des manuels couramment utilisés. Ces termes lui seront proposés de façon aléatoire (nous avons appelé ces programmes annexes loto 1, loto 2, etc) avec leur traduction

en français, et la description de certaines caractéristiques essentielles comme leur genre, leur accent tonique et le caractère fixe ou mobile de cet accent.

Mieux, avec le programme pourra être fourni un utilitaire de confection de lexiques bilingues, grâce auquel c'est l'utilisateur lui-même qui pourra se constituer sa propre base de données au fur et à mesure de son travail, en fonction de ses goûts et besoins. Ce fichier sera couplé à un programme d'extraction aléatoire de données qui fonctionnera de la même façon que les programmes annexes dont il vient d'être question.

Données techniques

Le matériel

Après avoir ainsi essayé de définir la conception générale du programme, nous nous sommes demandés sur quel type d'ordinateur il devait être réalisé et pouvoir "tourner". Et, à nouveau, nous sommes partis de la même question initiale: "à quel public devait s'adresser ce logiciel futur?".

Il s'agissait, à l'évidence, d'élèves ou d'étudiants, désireux d'acquérir ou de perfectionner certains automatismes dans l'utilisation de la déclinaison. Le sous-équipement de la plupart des établissements d'enseignement en matériel informatique en libre service et l'étonnante diversité de ce même matériel, le faible horaire attribué aux langues dans le second cycle du secondaire, ne laissaient entrevoir que peu de chances que ce futur logiciel puisse être utilisé régulièrement dans le cadre des horaires officiels de ces mêmes établissements. Et, par sa conception même, il supposait une interactivité immédiate machine/utilisateur qui n'était guère envisageable que dans le cas d'un emploi individuel hors du cadre scolaire traditionnel.

Il fallait donc un appareil individuel, le moins onéreux possible, peu encombrant, facile d'utilisation et qui permette néanmoins l'utilisation éventuelle d'une imprimante, une redéfinition des caractères au clavier et de larges possibilités de programmation. Parmi les ordinateurs dits "de jeux" qui apparaissaient comme les plus vendus sur le marché, le modèle **AMSTRAD CPC 6128** (depuis Octobre 1990, AMSTRAD CPC 6128 PLUS) nous a paru répondre à ces exigences. Nous avons, pour notre part, choisi le modèle équipé d'un moniteur couleur, qui nous offrait des possibilités intéressantes de mise en valeur de telle ou telle unité à l'écran par opposition de couleurs contrastées. Nous avons pu, par ailleurs, constater que les utilisateurs d'âge scolaire, familiarisés avec les jeux

informatiques, éprouvaient une répulsion certaine à l'égard des programmes présentés sur écran monochrome.

L'inconvénient majeur de cet appareil était, bien évidemment, sa relativement peu importante capacité de mémoire réellement disponible pour le programme, de l'ordre de 42 Ko. Pour pallier cet inconvénient, nous avons choisi de scinder notre programme en plusieurs modules spécialisés, dont chacun est capable de fonctionner de façon autonome.

Le programme complet, rédigé en BASIC, a donc été installé sur 4 disquettes double-face de format 3 pouces. La répartition des modules sur l'ensemble des disquettes est la suivante:

Disquette / Face	Module
1 A	1. le singulier des substantifs.
1 B	2. le pluriel des substantifs.
1 A	3. la séquence adjectif+substantif au singulier.
1 B	4. la séquence adjectif+substantif au pluriel.
1 A	5. la séquence pronom+substantif au singulier
1 B	6. la séquence pronom+substantif au pluriel.
2	7. les substantifs au cas par cas.
2 A	- le singulier.
2 B	- le pluriel
3 A	8. les noms de famille et patronymes.
3 B	9. le dictionnaire grammatical.
4	10. la séquence adjectif/substantif au cas par cas
4 A	- le singulier.
4 B	- le pluriel

NB. Le module 9 affiche sur le même écran à la fois les formes de singulier et de pluriel, mais n'est accessible qu'en simple consultation.

Redéfinition des caractères

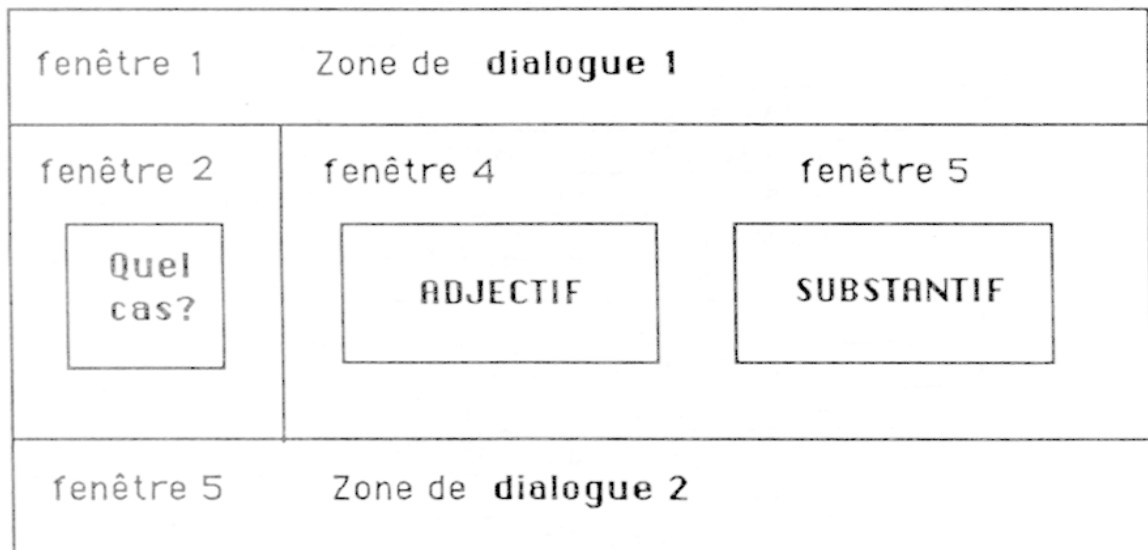
Le clavier utilisé étant un clavier AZERTY ordinaire, il a, bien évidemment, fallu l'adapter pour la saisie de données qui allaient être entrées en alphabet cyrillique. L'appareil retenu permettait de redéfinir les caractères grâce à une instruction spécialisée débrayable, et le programme de redéfinition a pu être conçu de façon à pouvoir revenir à l'alphabet latin, chaque fois que le besoin s'en ferait sentir.

Nous avons choisi une transposition des caractères qui se rapproche, le plus possible, d'une translittération pure et simple, accessible au non-spécialiste.

Fenêtrage de l'écran:

Un autre problème d'ordre technique se posait qui ne découlait pas du choix de l'appareil, mais du fait même que tout devait être affiché sur un écran dont les dimensions n'étaient pas extensibles. Nous avons voulu, dans un souci pédagogique, que les mêmes informations, les mêmes messages viennent s'afficher aux mêmes endroits, quel que soit le module en fonctionnement.

La disposition des fenêtres à l'écran et leur contenu ont donc été standardisés comme suit:



Tant que les termes proposés par l'utilisateur ne dépassent pas un nombre maximum de 11 signes, la disposition des fenêtres adoptée donne pleine satisfaction. Au dessus, l'adjectif ou le substantif risqueraient d'être décalés à la ligne suivante, et la *déclinaison complète* ne rentrerait plus dans le cadre qui lui a été réservé.

Nous avons donc fait jouer l'aptitude du programme à traiter des mots fictifs: il est demandé à l'utilisateur de redonner le même terme en le tronquant à partir de la gauche (perekvalifikacya >>>> -fikacja) sans, toutefois, descendre en dessous de 5 signes (= longueur maximale de certains suffixes pertinents), et le mot tronqué est traité comme si de rien n'était.

Cette situation tendant à devenir la norme dans le cas des formes de participes, en particulier de ceux qui présentent la particule - *sja*, nous

avons conçu une technique de travail spéciale qui permet de traiter isolément l'adjectif ou toute forme assimilée que l'utilisateur souhaite manipuler en l'absence d'un substantif de référence. Il suffit de proposer en tant que substantif une des formes du pronom personnel de 3-ème personne. La forme choisie (*on/onà/ono*) spécifie le genre selon lequel sera décliné l'adjectif, et joue en même temps le rôle d'un code spécifique qui ouvre à l'écran une fenêtre plus large, dans laquelle prendra place sans difficulté la déclinaison d'une forme aussi longue que *perekvalificirovavšijsja*

La démarche informatique

Le principe de base de fonctionnement du programme repose sur une constatation simple. Les diverses désinences casuelles sont interdépendantes, et peuvent, à quelques exceptions près, être déduites les unes des autres, par des processus de substitution élémentaires. Ainsi, au singulier, le modèle de référence sera-t-il le datif, où apparaît la totalité des modifications prévisibles du radical. Ce même datif singulier servira également de base pour la construction des formes de nominatif pluriel, les formes de datif, instrumental et prépositionnel pluriels étant fabriquées à partir de ce nominatif. La forme de génitif pluriel fera, pour sa part, l'objet d'un modèle particulier, relativement complexe.

Saisie au clavier

Le terme à traiter est proposé par l'utilisateur par saisie au clavier. S'il s'agit d'un substantif la forme de départ est, obligatoirement, la forme du dictionnaire, celle du nominatif singulier. Et, tout de suite, apparaît une difficulté. L'utilisateur débutant risque d'introduire par erreur une forme qui, en fait, n'est pas un singulier. L'expérience prouve que des mots comme *drova, perila* , par exemple, sont souvent interprétés par les élèves comme autant de formes de féminin singulier. Et le travail de notre premier filtre consistera, par référence à un fichier de données préétabli, à déceler leur statut de "pluralia tantum" et à le signaler à l'utilisateur par un message adéquat.

S'il s'agit d'un adjectif, la forme du dictionnaire pourra aussi, bien évidemment, être systématiquement proposée, quel que soit le genre du substantif qui intervient dans la même séquence. Il nous a paru, toutefois, très peu pédagogique d'exiger cette forme dans tous les cas, et nous avons fait en sorte que l'utilisateur puisse proposer n'importe quelle forme du nominatif de l'adjectif, masculine, féminine ou neutre. Dans ce cas l'ordinateur effectue une série d'opérations complexes qui visent dans un

premier temps à contrôler la graphie des désinences adjectivales en les corrigeant au besoin, puis dans un second temps à réduire la forme proposée à la forme de référence qui est celle du masculin. Après détermination du genre du substantif, cette forme de référence sera accordée et traitée en conséquence. L'ensemble des formes de nominatif aura préalablement été affiché à l'écran, selon le modèle *russky, aya, oc*

Ainsi, tout en permettant à l'utilisateur de proposer dès le départ une séquence accordée, l'ordinateur élimine toute possibilité d'erreur de sa part sur ce point.

Les possibilités de développement du programme

Ce programme a été présenté à de nombreux collègues et testé en situation sur des élèves du secondaire et du supérieur, qui tous lui ont réservé un accueil très favorable. L'efficacité en est certaine: le pourcentage d'erreurs survenant du fait du programme peut être considéré comme tendant vers zéro, et l'acquisition des automatismes n'étant plus ressentie comme un exercice scolaire, mais comme un jeu (il n'est pas rare que l'élève reste plus d'une heure au clavier sans même en prendre conscience!), les délais d'apprentissage sont considérablement raccourcis.

Mais surtout, par sa conception même, il permet d'envisager la création d'autres programmes encore plus ambitieux, dans lesquels l'utilisateur pourra faire entrer les termes de son choix dans des séries de structures syntaxiques répertoriées. C'est à dire qu'il sera possible de confectionner, entre autres, des exercices "à trous", dans lesquels la seule contrainte qu'aura à supporter l'utilisateur sera celle du contenu sémantique de l'ensemble de la phrase considérée, dont les éléments constituants pourront varier à l'infini à sa convenance.

Nous travaillons également à la réalisation d'un logiciel de même type "ouvert", portant sur le verbe et présentant la même souplesse. A partir de la seule forme d'infinitif, fournie librement par l'utilisateur, le programme pourra fournir, soit isolément, soit dans un environnement linguistique, toutes les formes de présent, de passé, de gérondifs et de participes du verbe considéré. Les résultats déjà obtenus sont très encourageants.

Et le couplage à venir des deux programmes (la déclinaison d'une part, la conjugaison au sens large de l'autre) ouvre des perspectives particulièrement prometteuses pour un enseignement du russe assisté par ordinateur.

Bibliographie

Ouvrages d'informatique.

AMSOFT, 1985, Amstrad CPC 6128. Manuel de l'utilisateur, AMSTRAD Consumer Electronics plc., 240 pages

ZARDETTO AKER, Sharon, 1985, Microsoft BASIC Programming for the Mac, Glenview /Illinois, Scott Foresman and Company, 344 pages

Microsoft Corporation, 1989, Microsoft QuickBASIC. Manuel de référence du langage BASIC, Microsoft Corporation, 568 pages

SZCZEPANOWSKI, 1984, Amstrad! Ouvre-toi!, Düsseldorf, Data Becker - Traduction française 1985 par Micro Application, 201 pages

KOWAL, 1986, Amstrad 6128. Le Grand livre du Basic, Düsseldorf, Data Becker - Traduction française 1986 par Micro Application, 263 pages

KAMPOW, 1986, Amstrad. Le Basic au bout des doigts, Düsseldorf, Data Becker - Traduction française 1985 par Micro Application, 189 pages

DELAUNOY Claude, 1983, Apprendre à programmer en Basic, Paris, Eyrolles, 254 pages

VIRGA, 1983, Programmer chez soi: le Basic, Belgique, Marabout, 244 pages

C. DE ROSSI, 1984, Apprentissage rapide du Basic, Paris, Eyrolles, 204 pages

Ouvrages grammaticaux de référence.

GARDE, Paul, 1980, Grammaire russe. Tome 1: Phonologie. Morphologie, Paris, Institut d'Etudes Slaves, 486 pages.

VINOGRADOV, V.V., 1960, Grammatika Russkogo @zxka. Tom 1: Fonetika i Morfologi@, Moskva, izdatelàstvo Akademii Nauk SSSR, 720 str.