

Conception et réalisation de logiciels : vers une plus grande intégration des connaissances de la langue

Louissette Emirkanian and Lorne H. Bouchard

Volume 16, Number 2, 1987

Problèmes linguistiques et enseignement du français au Québec

URI: <https://id.erudit.org/iderudit/602599ar>

DOI: <https://doi.org/10.7202/602599ar>

[See table of contents](#)

Publisher(s)

Université du Québec à Montréal

ISSN

0710-0167 (print)

1705-4591 (digital)

[Explore this journal](#)

Cite this article

Emirkanian, L. & Bouchard, L. H. (1987). Conception et réalisation de logiciels : vers une plus grande intégration des connaissances de la langue. *Revue québécoise de linguistique*, 16(2), 189–220. <https://doi.org/10.7202/602599ar>

Article abstract

The flexibility and the symbolic computing power of the microprocessor can be used to create a true interaction with the student and to support activities which enable the student to manipulate language and to reflect on its operation. We have tried to exploit these features in courseware the purpose of which is to improve the students' mastery of written French. The first program focusses on complex relative clauses and encourages the student to reflect explicitly on the way they function. The second program generates sentences containing a morphological, syntactic or semantic error and analyses sentences containing such errors.

CONCEPTION ET RÉALISATION DE LOGICIELS : VERS UNE PLUS GRANDE INTÉGRATION DES CONNAISSANCES DE LA LANGUE*

Louissette Émirkanian

Lorne H. Bouchard

0. Introduction

Depuis 1981, un nouvel outil a fait son apparition dans les classes du Québec : le micro-ordinateur. Ses utilisations sont multiples; de nombreuses classifications de ses applications pédagogiques ont été proposées (Dubuc (1982), Valentin (1983), Bestougeff et Fargette (1982)). L'ordinateur y est en général considéré comme un instrument de gestion pédagogique, un moyen d'enseignement, un objet d'apprentissage, etc. Nous nous intéresserons ici à l'ordinateur, outil d'apprentissage de l'écriture au primaire, plus particulièrement à l'ordinateur outil d'apprentissage de ce que le programme de français nomme «les connaissances et les techniques nécessaires à l'écriture» (Programme d'étude- Primaire (1979), p.27).

* Nos plus vifs remerciements vont à Benoît Tremblay, étudiant à la maîtrise en mathématiques et informatique (UQAM), qui participe activement à ce projet de recherche. Nous voudrions aussi remercier André Lacombe, titulaire de la classe de 6^e année (CECM-région nord) dans laquelle l'un des logiciels est expérimenté, et André Dugas pour leurs commentaires et leurs suggestions.

De nombreux logiciels à caractère pédagogique¹ ont vu le jour. La plupart d'entre eux utilisent le modèle d'enseignement directif (exercice répété, tutoriel); ils s'apparentent à ceux de l'enseignement programmé et en présentent tous les avantages et les inconvénients. De plus, il est frappant de constater que la majorité d'entre eux² portent sur le lexique (homophones, expressions figées, mots composés, antonymes, synonymes, etc.) ou la morphologie (orthographe grammaticale, participe passé, conjugaison, etc). Il est très tentant pour les enseignants du primaire d'utiliser dans leur classe de tels didacticiels étant donné que le programme de français, même s'il propose bon nombre de moyens pour développer l'habileté à écrire, reste assez discret sur le développement des connaissances, surtout celles qui ont trait à la syntaxe³.

La première question que l'on peut se poser est la suivante : ces logiciels portent-ils sur les difficultés réelles auxquelles se heurtent les enfants? La plupart du temps (nous parlons ici des logiciels commercialisés), ils abordent des notions que les enfants maîtrisent parfaitement ou des notions qui ne posent pas de réels problèmes dans la mesure où les enfants du primaire ne les emploient pas. On peut également se demander, et cela nous paraît essentiel, ce que l'utilisation de ces logiciels apporte de plus que celle d'un autre outil (fiches, manuels, etc.). Tout le monde s'accorde pour dire que le micro-ordinateur permet l'individualisation de l'apprentissage, l'initiative et l'interactivité. L'ordinateur est certes un médium interactif, c'est même là sa caractéristique essentielle, mais cela dépend évidemment avant tout du logiciel utilisé. Or, même si l'on ne s'attend pas (encore!) à trouver un dialogue enfant-machine, on constate que ces logiciels ne sont guère plus interactifs qu'un manuel. Prenons le cas de l'erreur. Si l'enfant se trompe, en règle générale, quand on n'affiche pas «Non, recommence», on propose les caractéristiques de la réponse attendue ou encore celles

1. Nous ne traiterons pas ici de l'utilisation des logiciels de traitement de texte, ni de celle du courrier électronique, outils qui peuvent être employés pour favoriser le développement de l'habileté à communiquer par écrit.

2. On pourra se reporter à l'article de Labelle (1986) sur les types de didacticiels existants pour l'enseignement du français.

3. Précisons que, jusqu'en 5e année, il n'est question dans la rubrique «code syntaxique» que de ponctuation. En 5e année, on parle de «manipulation de différentes structures de phrases» (p.260), mais il n'est question que du *ne* de la négation et de l'emploi fautif de certaines prépositions. En 6e année, on y ajoute l'emploi fautif de *que* à la place de *dont* (p.330).

de la réponse donnée par l'enfant. Les messages sont donc souvent les mêmes quelle que soit l'erreur commise. Il semble donc que toute la puissance de l'ordinateur n'est pas toujours exploitée. On s'attache souvent beaucoup plus aux aspects visuel et sonore (graphisme, couleur et musique par exemple) qu'au contenu.

Pour résumer, nous pourrions citer Dubuc (1985) qui énonce les critères de qualité d'un logiciel éducatif : «Tout d'abord, on demande d'un logiciel qu'il porte sur un problème d'apprentissage réel, pour lequel l'ordinateur est le médium le plus approprié.»

Il s'agit donc d'utiliser l'ordinateur pour ce qu'il a de spécifique : sa souplesse d'adaptation et sa puissance de calcul symbolique permettent d'une part de créer une véritable interaction avec l'enfant et d'autre part de favoriser des activités amenant l'enfant à manipuler la langue et à réfléchir sur son fonctionnement. C'est ce que nous allons essayer d'illustrer en présentant deux des logiciels que nous avons réalisés⁴. Ils s'inscrivent dans la tendance actuelle des logiciels basés sur la connaissance. À l'intérieur de ceux-ci, un système de calcul formel permet de représenter la connaissance de la langue et de la manipuler.

Nous nous sommes dans un premier temps appliqués à bâtir des didacticiels de français ayant pour but d'une part de pallier des problèmes précis auxquels les enfants se heurtent et d'autre part de combler une des lacunes des didacticiels de français existants : la pauvreté de la rétroaction.

Nous présenterons l'un de ces logiciels⁵. Il porte sur les relatives complexes. À la lumière des résultats des travaux effectués lors de cette première étape, nous essayons de bâtir un logiciel plus «ouvert» dans la mesure où il pourrait intervenir

4. Le premier a été réalisé dans le cadre d'une recherche subventionnée par la DGERU; le second, dans le cadre d'une recherche subventionnée en partie par le CRSHC et en partie par la DGERU.

5. Nous avons également réalisé un didacticiel dont l'objectif est de pallier la difficulté qu'ont les enfants du primaire à employer les clitiqes compléments et surtout les séquences de clitiqes (Émirkanian et Bouchard 1985b).

sur les connaissances de la langue en général. Ce domaine d'application est bien sûr beaucoup plus riche mais il pose de nombreux problèmes de formalisation.

Ce type de logiciel pourrait être utilisé dans le cadre de ce que Wertz (1980) appelle «l'apprentissage autonome». Nous le présenterons dans la deuxième partie de ce texte. Dans les deux cas, après avoir fait un survol des connaissances de la langue nécessaires que nous avons formalisées, nous présenterons à la fois la façon dont le didacticiel a été réalisé et celle dont il se déroule.

1. D'un didacticiel fermé...

S'ils n'éprouvent aucune difficulté avec les relatives en *qui* et *que*, les enfants du deuxième cycle du primaire ont de la difficulté à produire les relatives complexes (en *dont*, *sur laquelle*, etc.) sous une forme standard. Ils les produisent à partir de la 5^{ème} année sous une forme non-standard ou encore sous une forme agrammaticale (cf Dubuisson et Émirkanian 1982). Nous avons réalisé le didacticiel sur les relatives⁶ dans le but de favoriser chez l'enfant une réflexion explicite sur le fonctionnement du système des relatives.

1.1 Les connaissances de la langue

Il s'agissait dans un premier temps d'identifier et de formaliser les connaissances nécessaires à la réalisation de ce logiciel. Cette étape exige une grande rigueur. Contrairement à d'autres phénomènes, dans le cas des relatives, il n'y a pas toujours une seule réponse possible, mais il peut y en avoir plusieurs, et ce à différents niveaux. L'énoncé (1) peut être complété par les pronoms à *qui* ou à *laquelle* :

- (1) La seule personne ... ce chien obéisse est son maître.

6. Une version préliminaire de ce didacticiel a fait l'objet d'une communication à EAO84 (cf Émirkanian et Bouchard 1984).

Obéir est sous-catégorisé par un SP avec *à*. L'énoncé (2) peut être, quant à lui, complété par *contre qui*, *contre laquelle*, *avec qui* ou *avec laquelle* :

- (2) Annie ... je joue aux échecs va participer au prochain tournoi.

En effet, *jouer* peut, dans ce contexte, être sous-catégorisé par un complément SP avec les prépositions *contre* ou *avec*.

Dans le système formel, chaque pronom est caractérisé selon sa fonction (sujet ou complément), selon la ou les prépositions sous-jacentes s'il y a lieu, selon les traits Humain, Féminin, Singulier et Consonne. Les marqueurs de traits H, F, S et C sont suivis d'un des signes «+», «-» ou «*»; le «+» marque la présence d'un trait, le «-», la négation du trait et l'astérisque, la présence facultative du trait. Voici à titre d'exemple les traits retenus pour les pronoms *sur qui*, *à laquelle*, *qu'* et *dont* :

- | | | |
|-----|------------|--------------------------|
| (3) | SUR QUI | [COMP] [SUR] [H+ F* S*] |
| | À LAQUELLE | [COMP] [A] [H* F+ S+] |
| | QU' | [COMP] [] [H* F* S* C-] |
| | DONT | [COMP] [DE] [H* F* S*] |

Parallèlement à cela, nous avons compilé une base de données constituée de phrases contenant une relative et nous y avons organisé l'information de la façon suivante. Dans le cas de l'énoncé (2), par exemple, on note les éléments suivants :

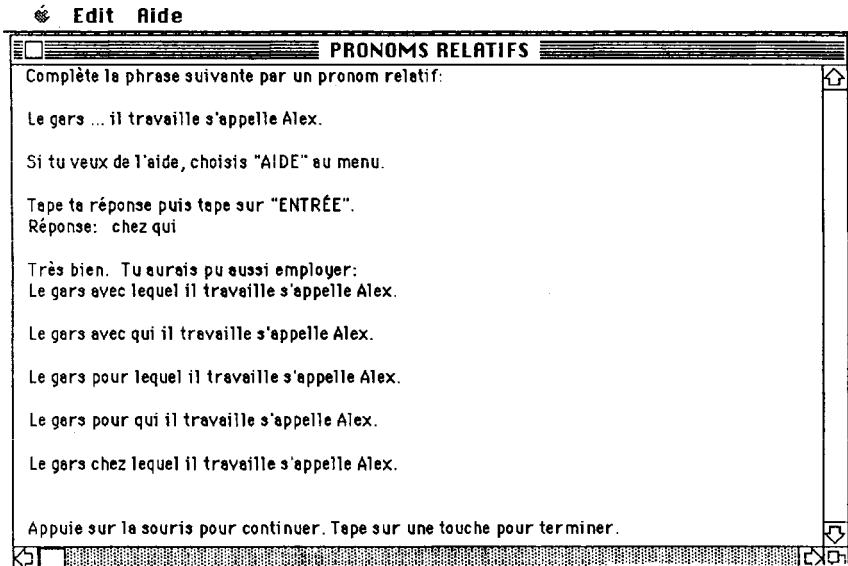
- | | | |
|-----|--------|---|
| (4) | P | [Annie ... je joue aux échecs va participer au prochain tournoi]. |
| | PREP | [avec contre] |
| | TRAITS | [H+ F+ S+] |
| | CATEG | [jouer *] |

On mentionne donc la ou les prépositions possibles, les traits de l'antécédent (*Annie* porte les traits +Humain, +Féminin et +Singulier); on note enfin le verbe avec la préposition requise ou un astérisque lorsque plusieurs prépositions sont admissibles.

1.2 La réalisation et le fonctionnement

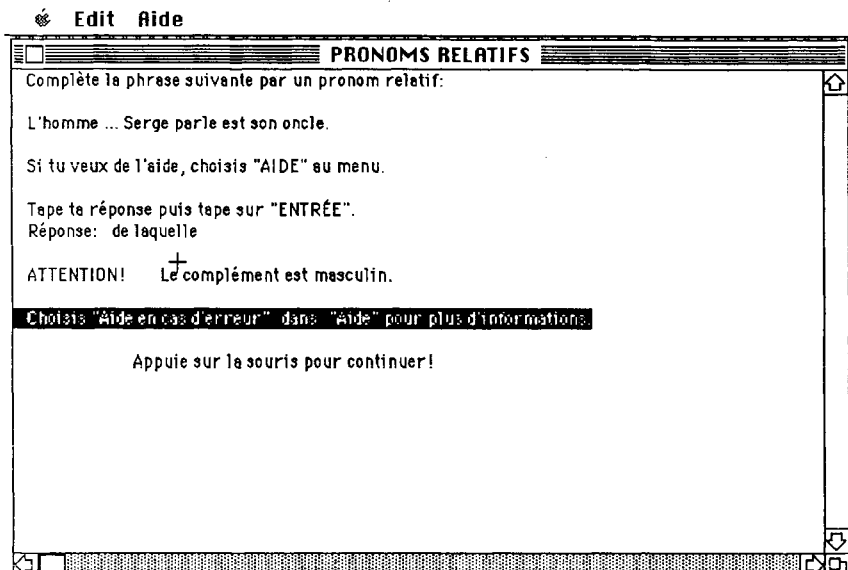
À partir de ces informations, le programme calcule automatiquement le ou les pronoms admissibles pour une phrase donnée. Que faire alors lorsque l'enfant

commet une erreur? L'un des buts de ce didacticiel a été de fournir une rétroaction pertinente et une aide précisément reliée à l'erreur. Si l'enfant propose une bonne réponse et que d'autres réponses sont possibles, le programme les lui présente. Pour l'énoncé présenté à l'écran 1, l'enfant propose *chez qui*; le programme lui présente les autres réponses possibles : *avec lequel, avec qui, pour lequel, pour qui et chez lequel*:

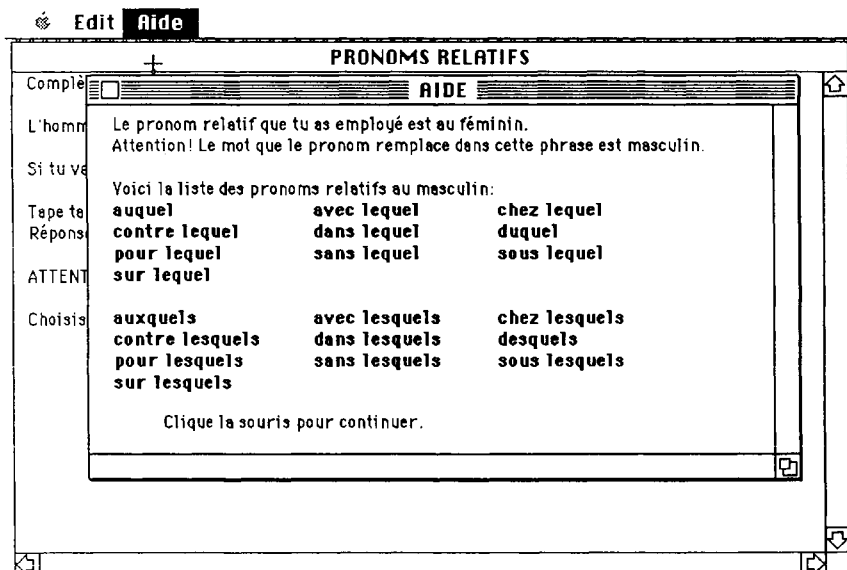


Écran 1

S'il y a erreur, le programme compare les caractéristiques de la réponse attendue à celles de la réponse proposée et génère automatiquement la rétroaction pertinente. Examinons quelques exemples. L'enfant fait une erreur d'accord en genre (écran 2); les caractéristiques de *de laquelle* [DE H* F+ S+] sont comparées à celles de la réponse attendue [[A DE AVEC] H+ F- S+]. La comparaison s'effectue de gauche à droite; la préposition *de* est ici admissible, le trait Humain également; le programme note, en revanche, la différence *F+ / F-* et génère automatiquement le message d'erreur et l'écran d'aide approprié (écran 3) est alors activé dans le menu «aide»; l'enfant peut le consulter s'il le désire :

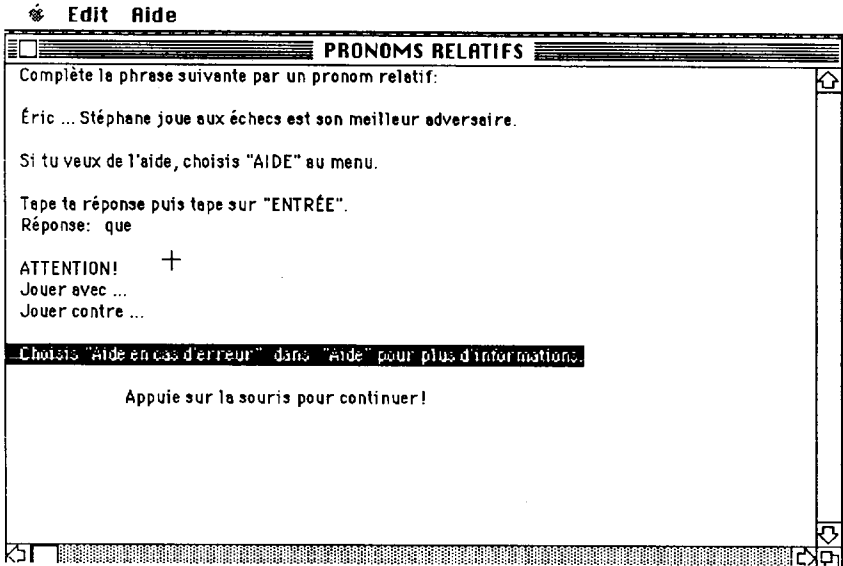


Écran 2



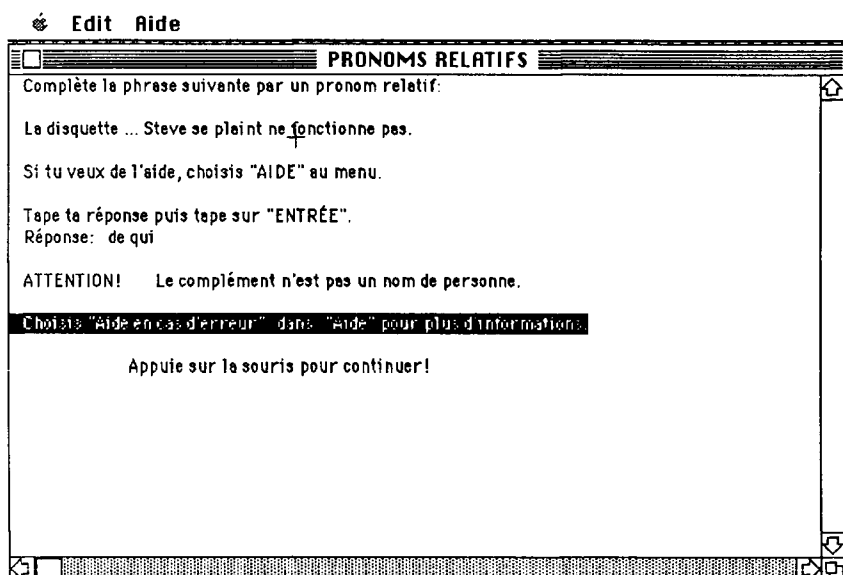
Écran 3

Pour ce qui est de l'erreur affichée à l'écran 4, la comparaison des traits est bloquée au niveau de la sous-catégorisation du verbe; le message d'erreur est automatiquement généré, il fait référence à la sous-catégorisation de *jouer* dans ce contexte :

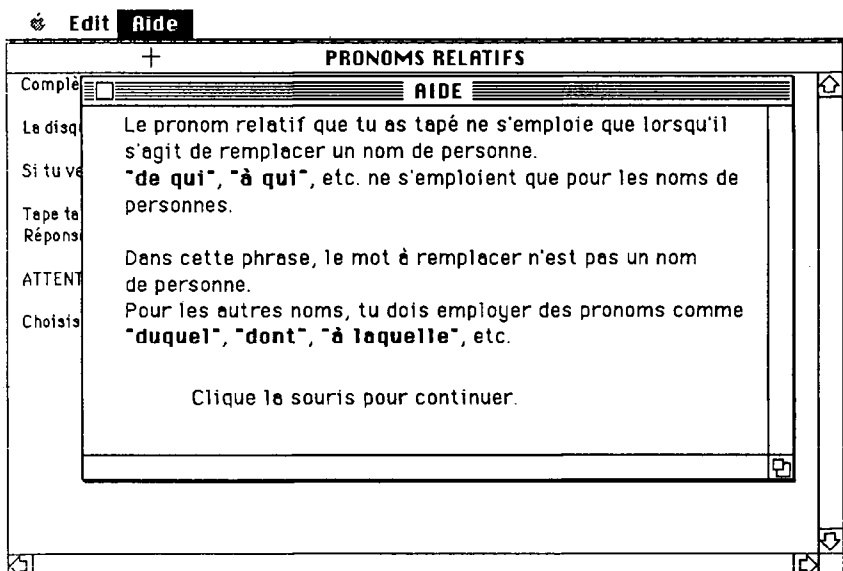


Écran 4

Dans l'exemple suivant (écran 5), *de qui* [DE H+ F* S*] est proposé. Les traits de la réponse attendue étant [DE H- F+ S+], le programme va noter l'opposition *H+/H-*. Le message d'erreur est automatiquement généré et l'enfant peut consulter l'écran d'aide (écran 6) qui devient alors disponible :



Écran 5



Écran 6

Ce type de didacticiel peut donc être utilisé pour remédier à un problème précis. Le domaine d'application d'un tel logiciel demeure réduit même si, à partir du système que nous venons de décrire, nous avons conçu et réalisé un autre didacticiel dont l'objectif est d'amener l'enfant à «découvrir» le fonctionnement du système des relatives (Bouchard et Emirkanian 1985a). Dans ces didacticiels, le dialogue personne/machine demeure extrêmement rigide et le modèle d'enseignement utilisé est encore celui de l'enseignement directif. Le logiciel dont nous allons parler maintenant pourrait déboucher, en revanche, sur un enseignement non-directif et favoriser une démarche exploratoire au sens de Papert (1980).

2. ... à un didacticiel ouvert

Le logiciel que nous avons réalisé permet d'une part de générer des phrases du français contenant au choix une erreur syntaxique, sémantique ou encore une erreur d'accord en genre ou en nombre et d'autre part d'analyser des phrases contenant des erreurs du même type. L'enfant doit, dans le premier cas, celui du programme *génération*, déceler l'erreur commise dans la phrase générée au hasard, dans le second cas, celui du programme *analyse*, il lui faut trouver l'erreur ou les erreurs commises par lui-même dans une phrase qu'il aura produite. L'objectif de ce logiciel est de rendre compte d'un sous-ensemble du français et non d'intervenir sur un problème précis. En effet, dans son état actuel, l'interface personne/machine est minimale. Il peut néanmoins servir de noyau à la réalisation de didacticiels portant par exemple sur des problèmes syntaxiques bien précis.

2.1 *Les connaissances de la langue*

Elles se trouvent formalisées dans la grammaire et le lexique. Nous avons programmé une grammaire déterministe très simple de type LALR(1) (cf Aho et

Ullman 1977). Le lexique⁷ est représenté par une base de données de mots auxquels sont associés des traits syntaxiques et sémantiques⁸ ainsi que des règles d'accord. Plutôt que de lister dans le lexique toutes les formes variantes d'un même élément (les noms, les verbes et les adjectifs, par exemple), nous avons uniquement noté la forme commune à laquelle nous avons greffé des règles permettant de réaliser l'accord, s'il y a lieu. En effet, nous n'avons pas à proprement parler isolé le radical du suffixe. Il s'agissait, surtout pour l'analyse, d'avoir l'élément non-commun aux deux formes (féminine et masculine, par exemple) composé du plus petit nombre de caractères possible. De plus, il est bien entendu qu'ici nous travaillons sur l'écrit et sur les variations graphiques et non phoniques. Les noms *chameau*, *chameaux*, *chamelle* et *chamelles* et les adjectifs *inquiet*, *inquiets*, *inquiète* et *inquiètes* sont notés de la façon suivante :

- | | | | |
|-----|-------|-----------------|------------------|
| (5) | chame | [Nom+ H- A+ C+] | [RG4 RN4 RN2] |
| | inqui | [Adj+ H+] | [RG11 RN2] [N_] |

Pour *chame-*, la règle de genre RG4 rend compte de l'opposition *-au/-lle* au masculin/féminin; la règle de nombre RN4, pour le masculin, rend compte de l'ajout d'un *-x* au pluriel et la règle de nombre RN2, pour le féminin, de celui d'un *-s*. Pour *inqui-*, la règle de genre RG11 rend compte de l'opposition *-et/ète*; la règle de nombre RN2 rend compte de l'ajout d'un *-s* au masculin et au féminin. Cet adjectif suit le nom. Pour le verbe, nous inscrivons le modèle de conjugaison auquel il appartient et sa sous-catégorisation; les verbes de la catégorie V7, par exemple, sont sous-catégorisés par un SN obligatoire et un SP facultatif. Prenons le cas du verbe *remarquer* en (6); il appartient à la catégorie V2, celle qui regroupe les verbes nécessitant la présence d'un syntagme nominal en position objet; cet élément peut porter le trait +Animé, Concret, Action, Qualité ou encore Sentiment; le sujet de *remarquer* doit être porteur du trait +Humain; sa conjugaison est de type 3 :

7. Une présentation plus élaborée du lexique a fait l'objet d'une communication à l'ACFAS (Émirkian et Bouchard 1986).

8. Certains traits sont indispensables pour la génération et l'analyse. D'autres ont été expressément notés pour la génération. En effet, si l'on décide de générer une erreur morphologique et que la phrase est par ailleurs sémantiquement bizarre, l'enfant pourrait s'arrêter surtout à l'aspect sémantique et ignorer l'erreur morphologique.

(6) remarquer

[V2

[SNs H+]

[SNo [+ A + CONCRET ACTION QUALITE SENTIMENT]]

[CONJ3]]

2.2 *La réalisation et le fonctionnement*

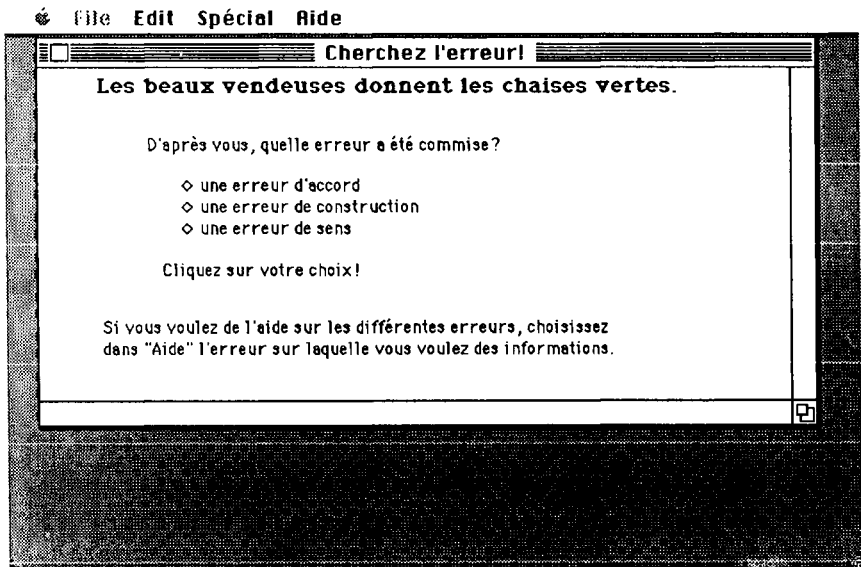
2.2.1 La génération

Nous avons élaboré un programme de génération qui permet d'engendrer des phrases d'un sous-ensemble du français contenant une seule erreur. La structure du programme de génération est typique des procédures de constructions sous contraintes (Émirkanian et Bouchard 1985a). Il s'agit de respecter les différentes règles sauf l'une d'elles; c'est cette violation qui générera l'erreur. La procédure principale choisit d'abord un verbe dans le lexique, construit ensuite le squelette de la phrase, choisit enfin le type d'erreur qui sera commis.

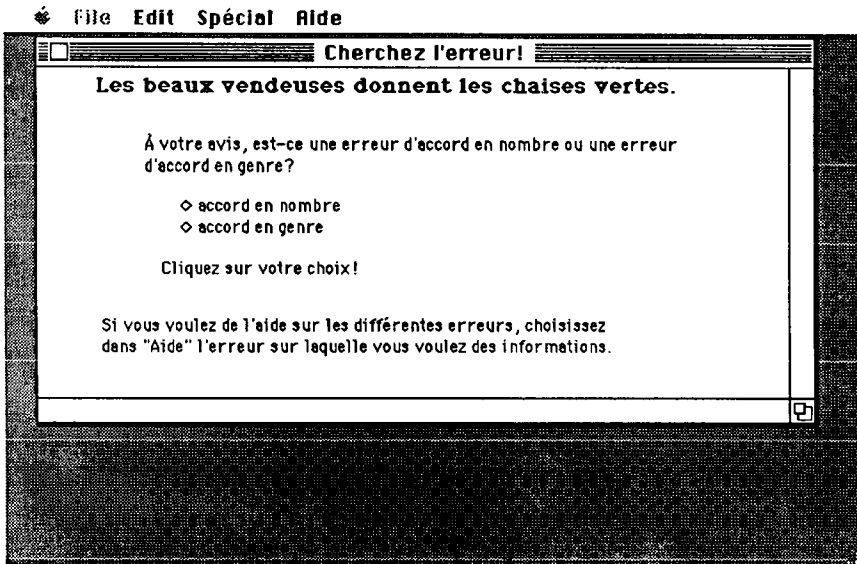
Nous illustrons les principales étapes de la génération des différents types d'erreur (erreur d'accord, erreur syntaxique, erreur sémantique).

Prenons le cas de la génération d'une erreur d'accord. Il s'agit de déterminer le type d'erreur. Il est choisi dans un éventail d'erreurs (accord en genre ou en nombre dans le *SNsujet*, le *SNobjet*, etc., accord en nombre entre *SNsujet* et verbe) en fonction du squelette de la phrase déjà fixé; si le squelette ne contient pas de SP, on ne pourra pas générer d'erreur d'accord dans un SP. Considérons le cas d'une erreur d'accord en genre dans le *SNsujet*. Il s'agit dans un premier temps de générer le groupe contenant l'erreur. La procédure de génération de ce groupe détermine les traits du syntagme qui s'ajoutent à ceux déjà déterminés par le verbe. Le syntagme sera par exemple [H+ F+ S-]. Puis, on choisit l'élément du syntagme sur lequel va porter l'erreur, l'adjectif par exemple; il portera le trait [F-]. Pour que l'erreur soit visible, le programme veille à ce que les éléments n'aient pas la même forme, à l'écrit, au masculin et au féminin. Cette vérification est facilitée par la structure que nous avons

adoptée pour le lexique, puisque chaque élément apparaît avec ses règles d'accord. Cette procédure génère automatiquement un message qui identifie précisément l'erreur commise. Ce message est alors stocké et sera utilisé par la suite pour indiquer l'erreur à l'enfant. La procédure de génération complète la phrase à partir du squelette sans commettre d'erreur. Les écrans 7 à 11 illustrent un cas d'erreur d'accord en genre dans le *SNSujet*. Dans le menu qui lui est proposé, l'enfant choisit bien «erreur d'accord» puis «erreur d'accord en genre» :

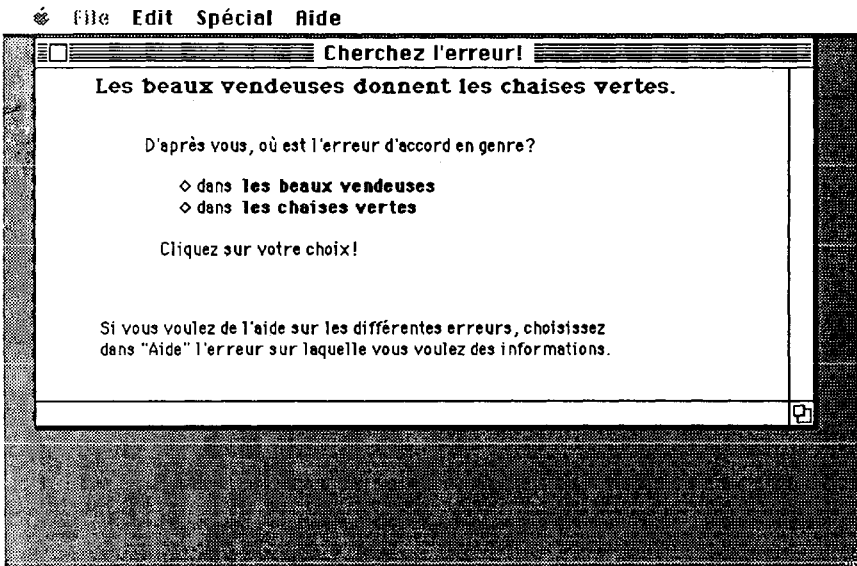


Écran 7

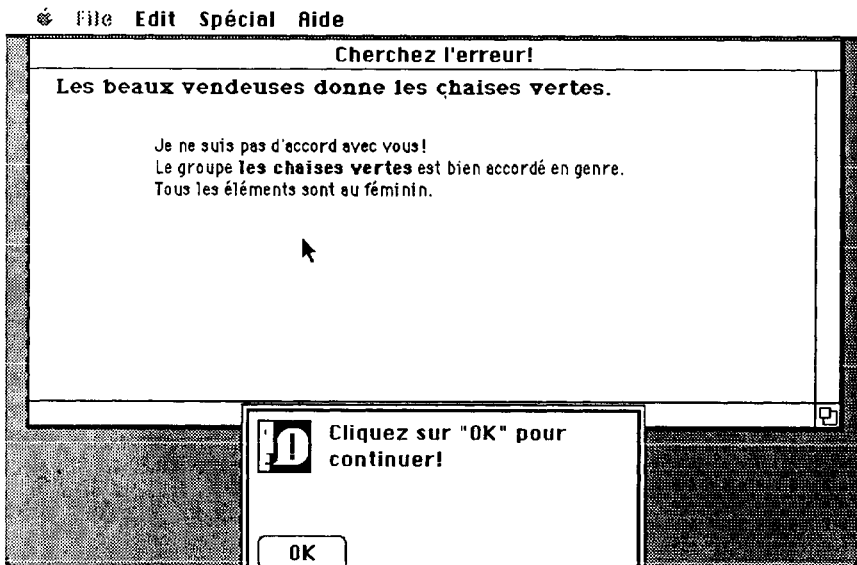


Écran 8

En revanche, il fait un mauvais choix quant au groupe dans lequel l'erreur a été effectivement commise. Il choisit *les chaises vertes*; le programme lui signale son erreur :

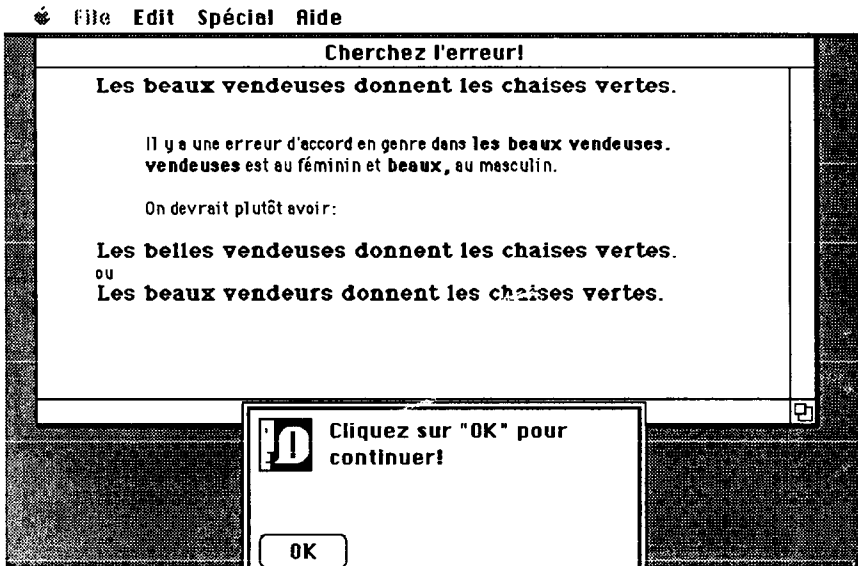


Écran 9



Écran 10

Il propose ensuite les deux énoncés possibles :



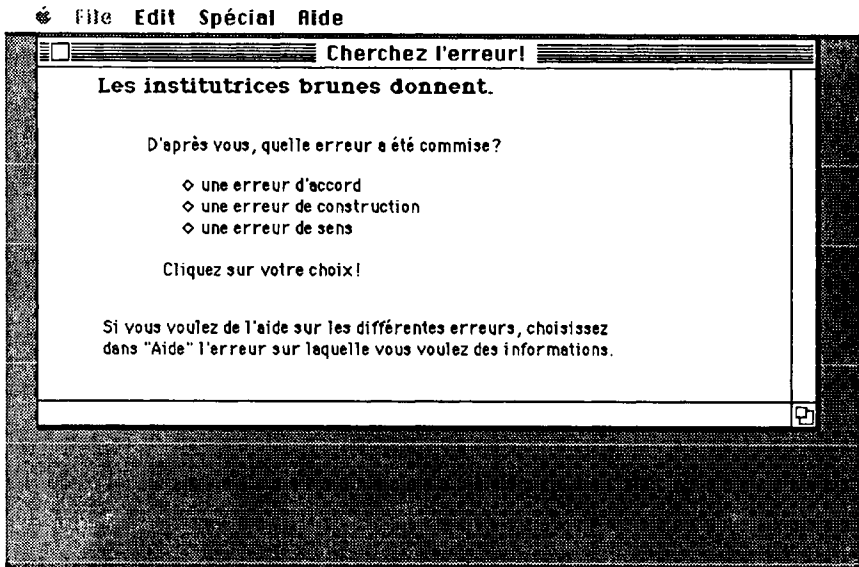
Écran 11

En effet, même si au départ, le programme génère l'erreur sur l'adjectif, au moment de donner la réponse, il tient compte de la possibilité d'avoir deux énoncés. S'il ne s'en était tenu qu'à ce qui avait été prévu au moment de la génération, il n'aurait pu proposer que *les belles vendeuses*; or, dans ce cas, *les beaux vendeurs* peut être aussi probable.

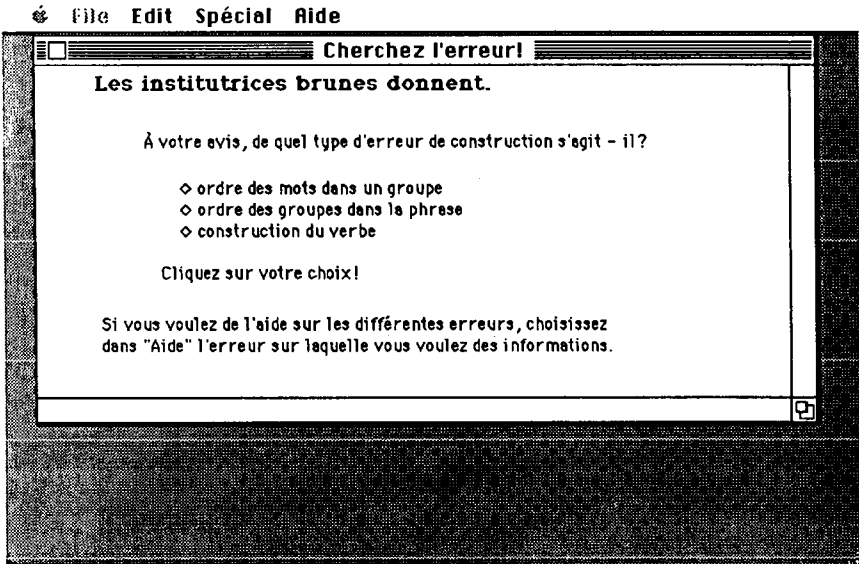
Examinons maintenant le cas de deux erreurs syntaxiques. On détermine à partir d'une liste d'erreurs syntaxiques (ordre des constituants, ordre des mots dans un constituant donné, sous-catégorisation du verbe) celles qui seront possibles avec le verbe choisi. Le programme en sélectionne une. Il s'agit, dans le premier exemple, de supprimer un élément qui sous-catégorise le verbe. Il suffit alors de générer la phrase en omettant simplement cet élément. Au niveau des erreurs syntaxiques, les contraintes sont nombreuses. En voici une, à titre d'exemple. Prenons le cas du verbe *compter* pour lequel nous avons prévu deux entrées lexicales : *compter1* (*compter sur*) où la présence du SP est indispensable et *compter2* (*compter des personnes, des choses*) où l'objet est facultatif. Supposons que le programme choisisse

de commettre une erreur au niveau de la sous-catégorisation de *compter*₁ et omette le SP; l'énoncé résultant sera tout à fait correct. On sera alors en présence de *compter*₂. Le programme tient compte de ces verbes qui ont plus d'une entrée lexicale.

Dans l'exemple ci-dessous, l'enfant choisit «erreur de construction» (écran 12), puis «construction du verbe» (écran13) dans les menus qui lui sont présentés :

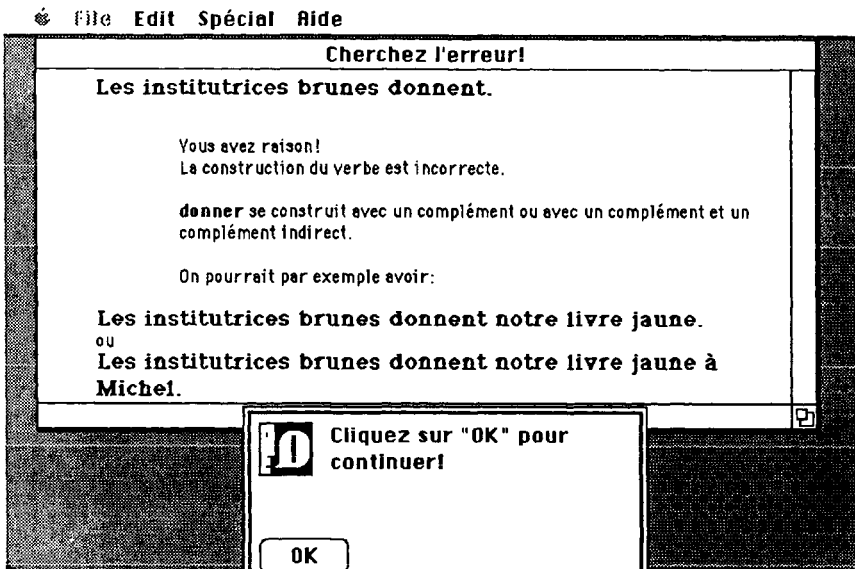


Écran 12



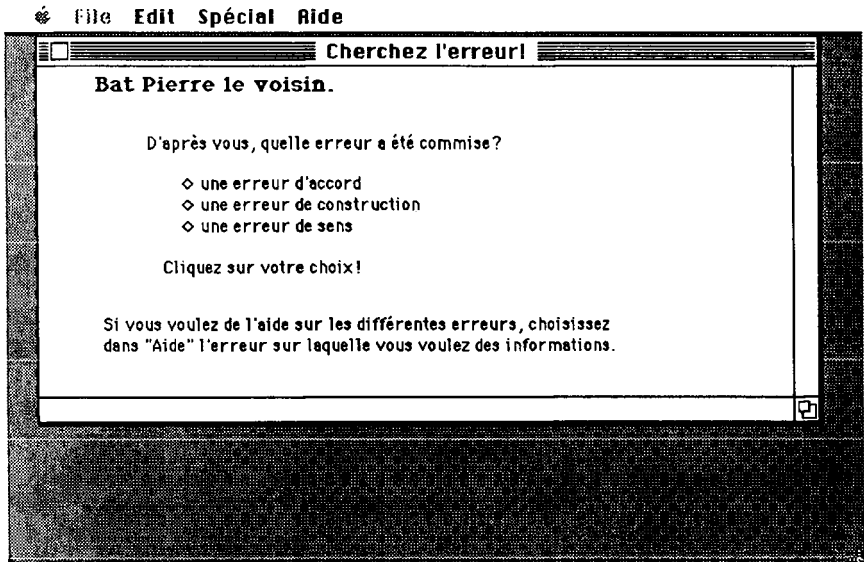
Écran 13

Le programme comme suite à la bonne réponse de l'enfant propose les deux constructions possibles de *donner* en allant chercher dans le lexique des éléments pouvant occuper la position laissée vacante lors de la génération de l'erreur (écran 14) :

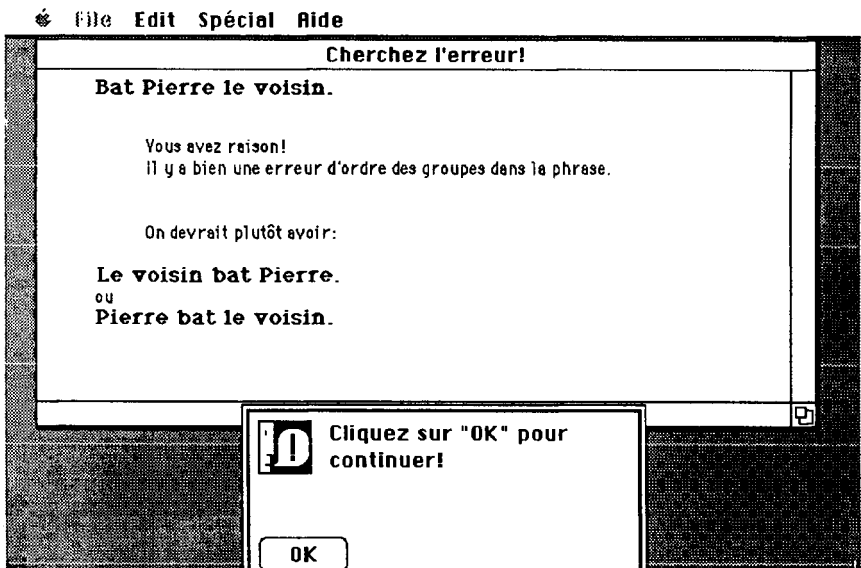


Écran 14

Dans le deuxième exemple, l'erreur porte sur l'ordre des constituants (écran 15). Que le programme ait généré l'erreur en déplaçant *Pierre, les voisins* ou encore *bat*, il prévoit la possibilité d'avoir, dans ce cas, deux phrases possibles (écran 16) :

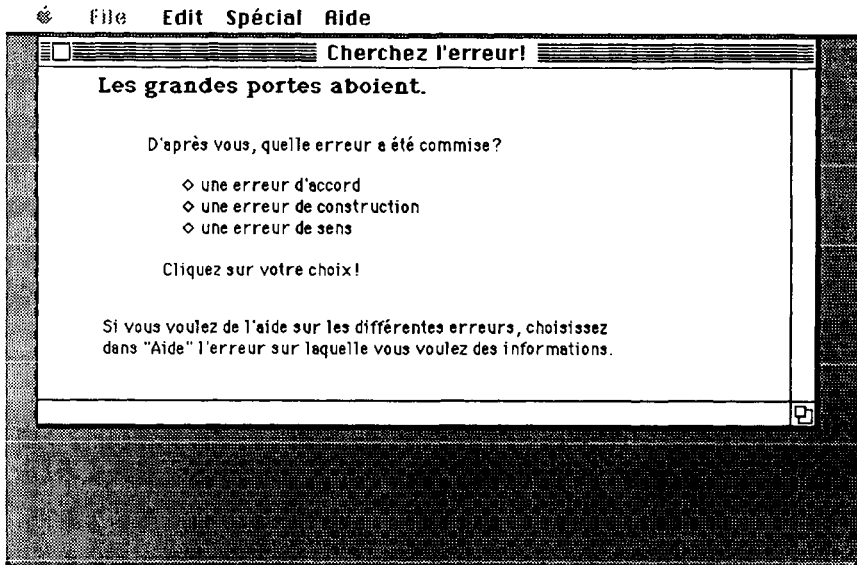


Écran 15

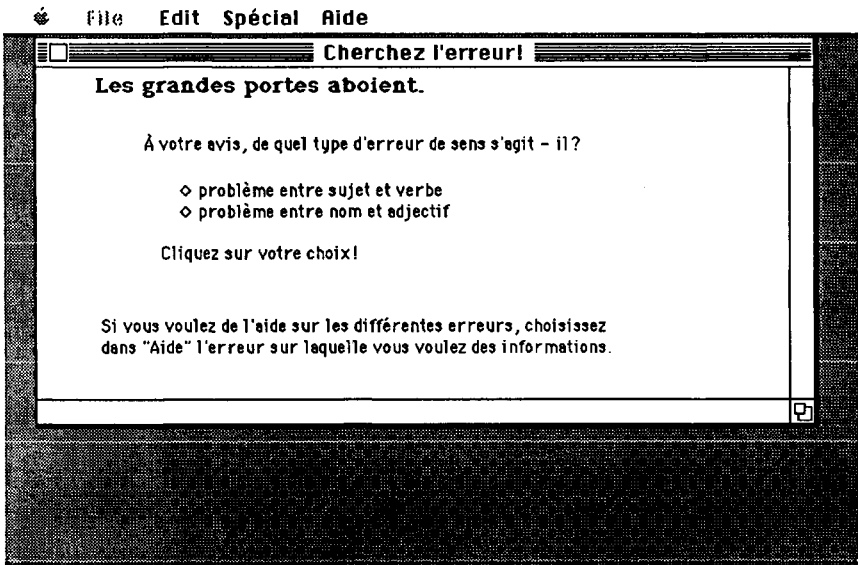


Écran 16

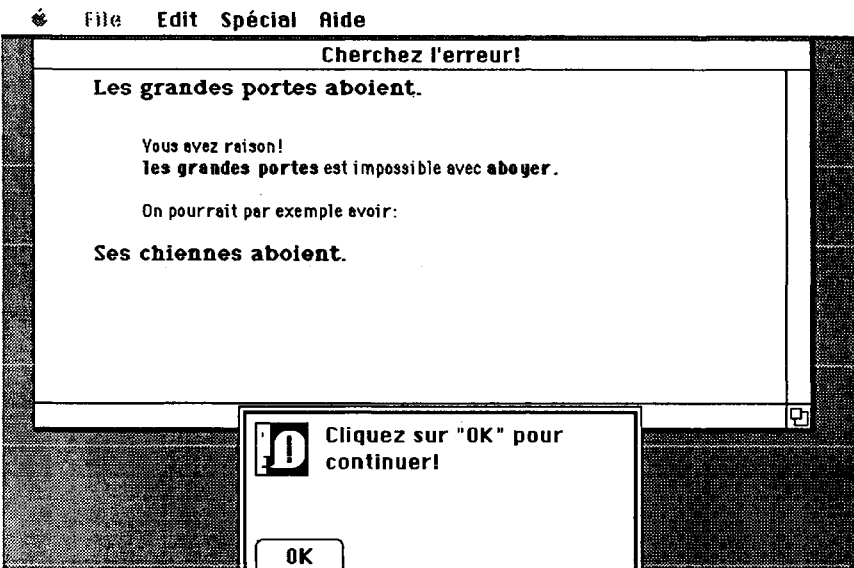
Pour ce qui est des erreurs sémantiques, on choisit dans la liste des erreurs (incompatibilités *SNsujet/verbe*, *nom/adjectif*, etc.) celles qui sont possibles avec le squelette de phrase. Le programme commet par exemple une erreur de compatibilité sémantique entre le *SNsujet* et le verbe. Il tient compte des traits déjà fixés, détermine au hasard les autres et produit le syntagme contenant l'erreur en prenant dans le lexique un nom qui n'est pas compatible avec les traits sémantiques du verbe. Il complète le *SNsujet* puis la phrase. Les écrans 17 à 19 illustrent la génération d'une erreur sémantique. L'enfant sélectionne «erreur sémantique» et la localise (problème entre le sujet et le verbe) :



Écran 17



Écran 18



Écran 19

Comme on peut le constater, le programme choisit *chiennes* dans le lexique.

En effet, pour certains verbes comme *aboyer* les contraintes peuvent être très rigides. Nous spécifions alors dans l'entrée lexicale les éléments appropriés; pour ce verbe, nous avons noté :

(7) Aboyer [V1
[SNs [= [chien [H- A+ C+] [RG14 RN2]]]]
[CONJ5]]

Dans le cas de la génération, quelle que soit l'erreur générée, le programme affiche des écrans de texte du type de ceux de 7 à 19. Le programme d'analyse, quant à lui, affiche graphiquement à l'écran les étapes de son fonctionnement.

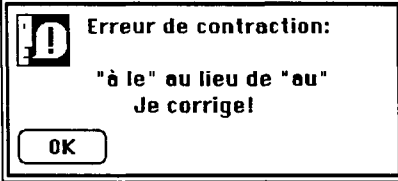
2. 2. 2 L'analyse

Afin de rendre la tâche d'analyse traitable, les enfants sont invités à construire une phrase à partir d'un menu de mots. Ce menu est déterminé de la façon suivante: le programme génère au hasard deux phrases correctes; les mots de ces phrases et leurs variantes morphologiques ainsi que quelques mots pris au hasard sont affichés.

L'analyse se déroule en trois grandes étapes. L'affichage graphique à l'écran permet de suivre pas à pas le déroulement du programme. La première étape porte sur les mots, la seconde construit l'arbre et la troisième détecte les erreurs en cheminant dans l'arbre.

La première étape comprend trois phases : détection et correction des erreurs de contraction et d'élision, expansion et enfin catégorisation des mots. Lors de cette étape, les deux ou trois premiers niveaux (s'il y a eu expansion) sont affichés au bas de l'écran. La phase de détection et de correction des erreurs de contraction et d'élision vérifie si l'on a des séquences du type *à le*, *de le* ou encore *le oiseau* et les corrige immédiatement (écrans 20 et 21) :

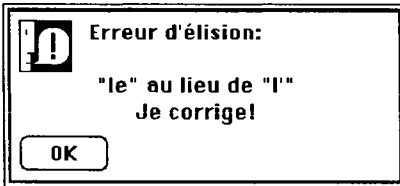
File Edit Debug



Jean offre un stylo à le frère

Écran 20

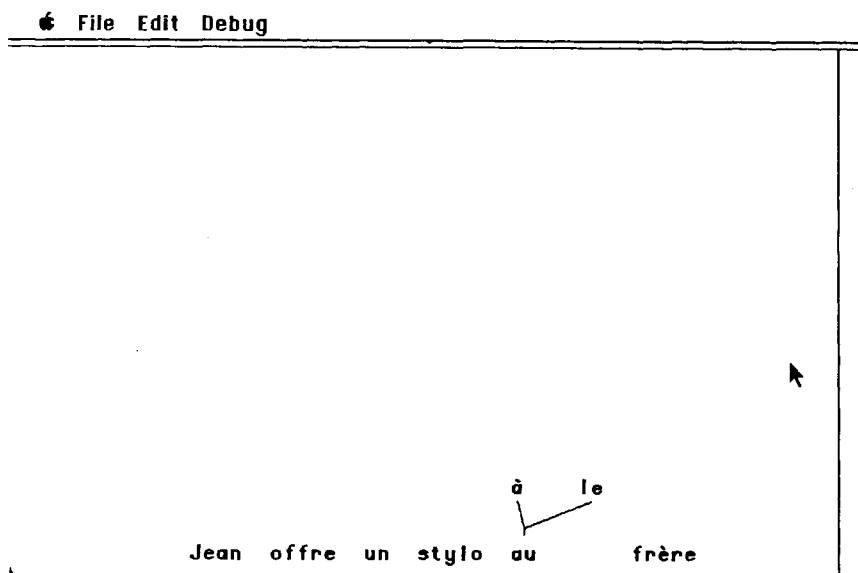
File Edit Debug



le enfant regarde le gros chien triste

Écran 21

La deuxième phase décompose les mots contractés en leurs éléments primitifs : par exemple *au* devient *à le* (écran 22) :

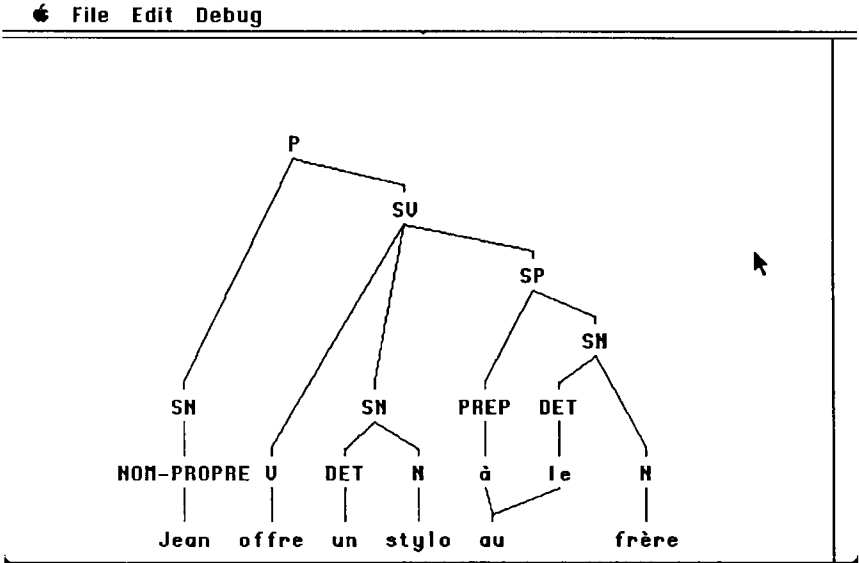


Écran 22

La troisième phase, une des plus importantes, caractérise les éléments en allant consulter le lexique⁹. Pour les noms, les adjectifs et les verbes, il est fort probable que l'on ne trouve pas la forme telle quelle dans le lexique. Le programme consulte alors la liste des suffixes, ceux du nombre et ceux du genre, ou ceux des conjugaisons. Prenons le cas du mot *rédactrice* ; cette suite de caractères ne figure pas dans le lexique. Le programme consulte la liste des suffixes du nombre, *-e* ne s'y trouve pas; en revanche, il est présent dans la liste des suffixes du genre; néanmoins, *rédactric-* ne se trouve toujours pas dans le lexique; étant donné qu'il n'y a ni suffixe *-ce*, ni suffixe *-ice*, mais un suffixe *-rice*, le programme soustrait *-rice* et trouve *rédact.*

9. Nous bâtissons actuellement, dans le cadre d'une autre recherche, un algorithme de proximité lexicale capable de détecter les erreurs d'orthographe les plus fréquentes.

C'est lors de la seconde étape que s'applique l'analyseur ascendant qui termine la construction de l'arbre et l'affiche simultanément à l'écran (écran 23) :



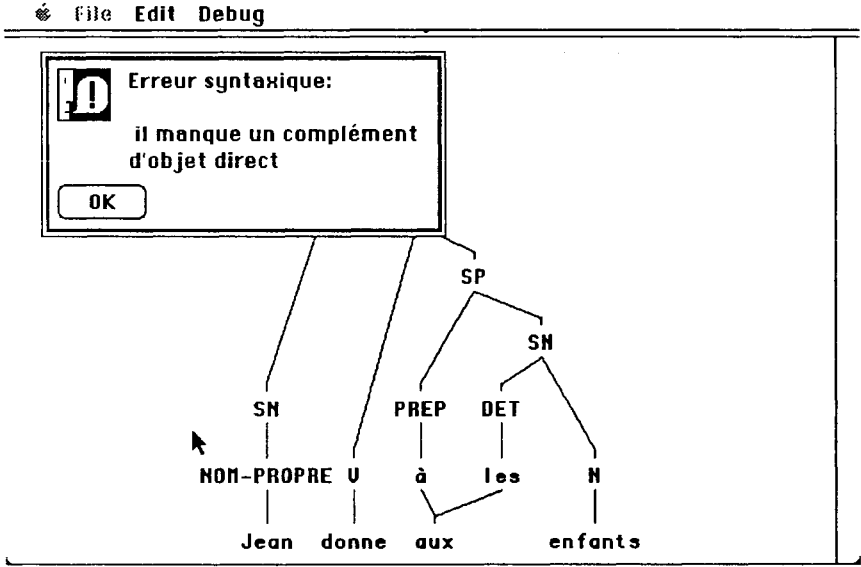
Écran 23

Dans certains cas, par exemple celui d'une erreur dans l'ordre des constituants majeurs, l'analyse est bloquée avant la dérivation de P. Le programme applique alors une grammaire qui caractérise les erreurs de ce type; associés à chacune de ces règles d'erreur se trouvent un diagnostic et une suggestion de correction possible.

La troisième étape comprend trois phases : la suite de l'analyse syntaxique et les analyses morphologique et sémantique. Elles s'effectuent à partir de l'arbre, des informations recueillies dans le lexique au moment de l'analyse lexicale et des règles d'accord et de compatibilités sémantiques. Durant cette étape, l'arbre reste affiché à l'écran et le programme le parcourt lors des différentes analyses.

Examinons un type d'erreur syntaxique. Pour la phrase **Jean donne aux enfants* (écran 24), la deuxième étape a donc été franchie sans incident puisque le programme a pu construire un arbre. La suite de l'analyse syntaxique est donc effectuée.

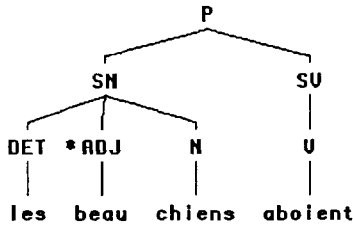
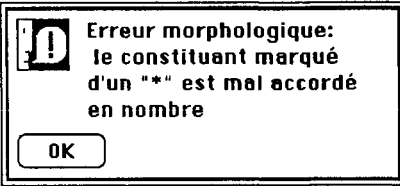
Le programme analyse la structure du syntagme verbal à la lumière des traits de sous-catégorisation de *donner* (verbe de la catégorie V7, SN obligatoire, SP facultatif) et observe que le SN est absent. Le message d'erreur est généré :



Écran 24

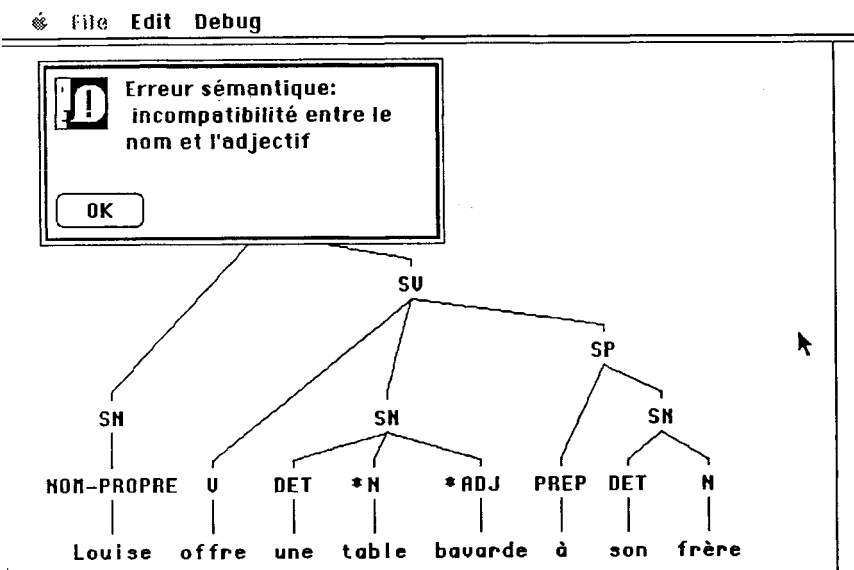
La seconde phase détecte les erreurs d'accord. Le programme isole les syntagmes nominaux et vérifie que toutes les règles d'accord en genre et en nombre ont été appliquées; il vérifie ensuite l'accord entre le SN _{sujet} et le verbe. L'écran 25 illustre la détection d'une erreur d'accord en nombre de l'adjectif. Notons que la position dans l'arbre de l'erreur est marquée d'un astérisque :

File Edit Debug



Écran 25

La troisième phase détecte les erreurs sémantiques. Le programme vérifie d'abord la compatibilité dans les syntagmes nominaux des noms et des adjectifs (s'il y a lieu), ensuite celle du *SNsujet* et du verbe puis, s'ils sont présents, celles du *SNobjet* et du *SP* et du verbe. L'écran 26 illustre la détection d'une erreur de compatibilité entre le nom et l'adjectif dans le groupe *SNobjet* :



3. Conclusion

L'enseignement intelligemment assisté par ordinateur (EIAO) (Carbonell 1970) est une évolution naturelle de l'enseignement assisté par ordinateur (EAO) rendue pratiquement possible par l'augmentation de la capacité de stockage et celle de la puissance de calcul du micro-ordinateur. Ce qui caractérise l'EIAO par rapport à l'EAO, c'est essentiellement l'utilisation des techniques de l'intelligence artificielle, en particulier le calcul symbolique (Newell et Simon 1981), pour représenter d'une part la connaissance du domaine à enseigner (Brown et al. 1982) et modéliser d'autre part l'état de la connaissance chez l'apprenant (Brown et Burton 1978). Les didacticiels que nous avons présentés ont été réalisés en Logo (Harvey 1985), un environnement de programmation agréable pour le calcul symbolique. Le didacticiel sur les relatives a d'abord été réalisé sur un Apple IIe, malgré la taille de mémoire (128Ko) et la capacité de calcul limitées de cet appareil. Ce didacticiel a ensuite été transféré sur un Macintosh (512Ko). Le second didacticiel, en particulier la *génération*, exploite à fond les possibilités de ce micro-ordinateur. Il sera expérimenté sous peu. Lors de

cette expérimentation, nous avons l'intention de recueillir systématiquement des données sur les erreurs commises par les enfants. Une analyse des données, éclairée par des interviews avec les enfants, nous aidera à mettre au point un programme qui construit un modèle de la connaissance et qui permet d'ajuster le comportement du didacticiel au problème précis de l'apprenant.

Louissette Émirkanian
Département de linguistique

Lorne H. Bouchard
Département de mathématiques et d'informatique

Université du Québec à Montréal

Références

- AHO, A. V. et J. D. Ullman (1977) *Principles of Compiler Design*, Addison-Wesley.
- BESTOUGEFF, H. et J. P. Fargette (1982) *Enseignement et ordinateur*, Cedic/Fernand Nathan, 185p.
- BOUCHARD, L. H. et L. Émirkanian (1985) «A Formal System for the Relative Clauses in French and its Uses for CAL», *Proceeding COMPINT85*, p.32-34.
- BROWN, J. S. et al. (1982) «Pedagogical, natural language and knowledge engineering techniques in SOPHIE I, II and III» dans D. Sleeman et J. S. Brown (réd.), *Intelligent Tutoring Systems*, Academic Press, pp.227-282.
- BROWN, J. S. et R. R. Burton (1978) «Diagnostic Models of Procedural Bugs in Basic Mathematical Skills», *Cognitive Science*, Volume 2, pp.155-192.
- CARBONELL, J. R. (1970) «AI in CAI : An Artificial Intelligence Approach to Computer-Aided Instruction», *IEEE Trans. on Man-Machine Systems*, Volume11, pp.190-202.
- DUBUC, L. (1982) *Classification des applications pédagogiques de l'ordinateur*, Ministère de l'éducation, DGME, Gouvernement du Québec, 9p.
- DUBUC, L. (1985) «Normes de qualité du logiciel éducatif», *Bip Bip*, n°37, pp.41-44.
- DUBUISSON, C. et L. Émirkanian (1982) «Acquisition des relatives et implications pédagogiques», dans C. Lefebvre (réd.), *La syntaxe comparée du français standard et populaire : approches formelle et fonctionnelle*, Office de la langue française, Gouvernement du Québec, pp.367-397.
- ÉMIRKANIAN, L. et L. H. Bouchard (1984) «Utilisation de l'ordinateur pour l'apprentissage de la langue», *Actes du premier colloque scientifique francophone sur l'enseignement assisté par ordinateur EAO84*, Agence de l'Informatique, pp.98-110.
- ÉMIRKANIAN, L. et L. H. Bouchard (1985a) «Un exemple d'exploration linguistique du français à l'aide de Logo», *Revue Québécoise de linguistique*, Volume 14, n°2, pp.145-156.
- ÉMIRKANIAN, L. et L. H. Bouchard (1985b) «Conception et réalisation d'un didacticiel sur les pronoms personnels», *Bulletin de l'APOP*, Volume III, n°3, pp.10-13.
- ÉMIRKANIAN, L. et L. H. Bouchard (1986) *Analyse et génération de phrases d'un sous-ensemble du français*, Communication, 54e congrès de l'ACFAS, Université de Montréal.

- HARVEY, B. (1985) *Computer Science Logo Style*, The MIT Press, 319p.
- LABELLE, F. (1986) «Des défis pour la linguistique appliquée», *Bulletin de l'ACLA*, Volume 8, n°1, pp.23-31.
- NEWELL, A. et H. A. Simon (1981) «Computer Science as Empirical Enquiry : Symbols and Search» dans J. Haugeland (réd.), *Mind Design*, MIT Press, pp.35-66.
- PAPERT, S. (1980) *Mindstorms. Children, Computers, and Powerful Ideas*, Basic Books, Inc., 230p.
- Programme d'étude-Primaire, Français* (1979) Direction des programmes, Gouvernement du Québec, 334p.
- VALENTIN, D. (1983) «Un ordinateur? pour quoi faire?», *Le français aujourd'hui*, n°63, pp.47-56.
- WERTZ, H. (1980) «L'utilisation de l'outil informatique : pilotage et programmation» dans J. Cl. Simon, *L'éducation et l'informatisation de la société (Annexes)*, La Documentation Française, pp.177-194.