

QUEEN
TK
7882
.I6
T4614 #5

C-2.

IC

TÉLIDON

RECHERCHE EN COMPORTEMENT SUR TÉLIDON

5 Langages d'interrogation interactive pour bases de données externes

2 / LANGAGES D'INTERROGATION INTERACTIVE

POUR

BASES DE DONNÉES EXTERNES /

/ F.H. Lochovsky, /

D.C. Tschritzis

Mars 1983

Industry Canada
LIBRARY
SEP 14 1998
BIBLIOTHEQUE
Industrie Canada

Groupe de recherche sur les systèmes informatiques
Université de Toronto
Toronto (Ontario) M5S 1A1
Canada

Pour: le ministère des Communications, Ottawa, Canada
En vertu du contrat passé avec le ministère
des Approvisionnements et Services, numéro d'ordre OSU80-00124
Période: du 1er juillet 1980 au 31 mars 1981

COMMUNICATIONS CANADA
JUL 9 1983
LIBRARY - BIBLIOTHEQUE

Des exemplaires de ce document
sont disponibles gratuitement auprès de la:

Direction générale de l'information
Ministère des Communications
300, rue Slater
Ottawa (Ontario)
K1A 0C8

No. DOC-TBR-DBRE-81-5-F

DD 375 3141
DL 375 9419

TK
7892
I6
T46f
#5

RÉSUMÉ

La quantité de données informatisées disponibles s'accroît sans cesse. Une bonne partie de ces données sont informationnelles ou destinées à la consultation et elles pourraient éventuellement être mises à la disposition du grand public. Jusqu'à récemment, il n'existait pas de support commode qui permette au public d'avoir accès à cette information. Cependant, deux événements, soit la croissance de réseaux de communication et l'arrivée sur le marché des systèmes vidéotex, offrent maintenant au public la possibilité d'accéder facilement à cette information. Dans le présent rapport, nous examinerons la nature du mécanisme qui permet, depuis le foyer et grâce à un service d'information vidéotex, d'avoir accès à des bases de données externes au système vidéotex proprement dit.

Nous étudierons d'abord le processus d'accès à une base de données (interrogation), d'après les caractéristiques de l'interaction du point de vue de l'utilisateur; (dans les pages qui suivent, nous parlerons de l'"interaction de l'utilisateur"). Nous divisons le processus de l'interaction en trois parties: la demande, la réponse et la dynamique. Pour les trois parties, certaines caractéristiques qui décrivent les paramètres de l'interaction de l'utilisateur, sont identifiées. Dans le cas des demandes, les paramètres sont le nombre de frappes requises, le type et le nombre de commandes disponibles, le degré de liberté quant à la formulation de la demande, la sélectivité des demandes, leur uniformité par rapport aux bases de données auxquelles on a accès et la possibilité de personnaliser la formulation de la demande en fonction d'applications précises. En ce qui a trait aux réponses, les paramètres sont la forme dans laquelle la réponse est présentée à l'utilisateur, la question de savoir si la réponse peut être présentée par l'intermédiaire de plus d'un support, si la réponse peut être personnalisée, le degré de maîtrise dynamique sur la réponse et la possibilité de conserver la réponse et de l'utiliser plus tard comme entrée. Pour ce qui est de la dynamique de l'interaction, les paramètres concernent l'intensité de l'interaction entre l'utilisateur et le système, le degré de ludicité que comporte l'interaction, le nombre de protocoles disponibles pour l'interaction, la souplesse du système lorsque l'utilisateur fait une demande et le degré de commande dont il bénéficie dans une interaction. Pour chacun de ces paramètres, une "valeur" souhaitable est déterminée.

Nous isolons ensuite sept types de langages d'interrogation: le langage à mots-clés, par exemples, naturel, à menus, à graphiques, à supports multiples et à jeux informatiques. Chaque genre de langage d'interrogation est décrit et évalué en fonction des paramètres d'interaction.

Il ressort de cette évaluation des langages d'interrogation qu'aucun genre de langage ne réunit à lui seul toutes les caractéristiques souhaitables. Nous proposons par conséquent certaines exigences auxquelles devrait répondre un langage d'interrogation pour les bases de données externes. Compte tenu de ces exigences, nous proposons ensuite un modèle de langage d'interrogation qui donne accès aux bases de données externes.

Pour décrire un langage d'interrogation, un paradigme représentant un vaisseau spatial personnel naviguant dans l'espace à l'aide de cartes détaillées est utilisé. Essentiellement, l'utilisateur peut demander des cartes de l'espace informationnel parmi celles qui sont à sa disposition et choisir sa destination à partir de ces cartes. Celles-ci sont présentées sous la forme de modèles qui représentent la structure et(ou) le contenu des bases de données externes. La plus grande partie de l'interaction est accomplie en pointant les zones de modèles et en entrant les valeurs ou en demandant l'affichage de ces dernières. La nature de l'interaction permet de manipuler de manière uniforme les données mises en forme, les données d'un texte non mises en forme, les données sous forme d'image et les données sous forme vocale.

TABLE DES MATIÈRES

1	INTRODUCTION	1
2	PARAMÈTRES D'INTERACTION DE L'UTILISATEUR	3
	2.1 Demande	5
	2.2 Réponse	8
	2.3 Dynamique	10
	2.4 Résumé	13
3	ÉVALUATION DU LANGAGE D'INTERROGATION	15
	3.1 Langages à mots-clés	15
	3.2 Langages par exemples	17
	3.3 Langage naturel	20
	3.4 Menu	22
	3.5 Graphiques	24
	3.6 Supports multiples	26
	3.7 Jeux informatiques	29
4	BESOINS EN MATIÈRE D'INTERROGATION	33
5	PARADIGME	39
6	VUE D'ENSEMBLE DE LA CONCEPTION	43
	6.1 Choix d'une base de données externe	43
	6.2 Exploration des données	51
	6.3 Itinéraires préétablis	53
7	CONSIDÉRATIONS EN MATIÈRE DE CONCEPTION	55
	7.1 Pointage	55
	7.2 Positionnement dans le portique	57
	7.3 Fonctions de commande	60
	7.4 Modes de spécification	63
	7.5 Types de données	65
8	REMARQUES ET CONCLUSIONS	69
	BIBLIOGRAPHIE	75

1 INTRODUCTION

Au cours des dernières années, on a assisté à une croissance marquée du nombre des sources d'information informatisées (bases de données) auxquelles on a accès en direct. Beaucoup de ces bases de données en liaison directe fournissent de l'information au public. C'est le cas, par exemple, des systèmes des bibliothèques, des transporteurs aériens et des services d'information des gouvernements. Très souvent cependant, le public n'a pas d'accès direct et facile à ces bases de données. Un intermédiaire interroge la base de données et transmet la réponse ou encore, l'information est donnée de façon non sélective dans des endroits précis.

Une autre réalisation des dernières années a été l'installation et l'expansion des réseaux de communication. Ces réseaux permettent d'interroger les bases de données à distance et à un coût raisonnable. Il devient ainsi possible au public d'accéder en direct aux bases de données "d'information destinée au consommateur". Mais une question se pose: comment et de quel endroit se fera cet accès?

Ces années ont aussi été marquées par l'arrivée des systèmes vidéotex. Il s'agit de systèmes interactifs de communication visuelle dont l'un des buts est de permettre au public d'accéder à des bases de données externes. (Par base de données externe, on désigne une base de données dont le contenu et la structure ne relèvent pas du système de livraison de l'information (vidéotex).) Télidon est l'un de ces systèmes (Bown et autres, 1978). L'interaction avec le système se fait par l'intermédiaire d'un téléviseur modifié, qui sert de terminal de visualisation, et d'un clavier ou d'un bloc de touches, qui est utilisé pour choisir les données à afficher.

Langages d'interrogation interactive pour bases de données externes

Comme le public aura un jour accès aux systèmes vidéotex, l'interaction de l'utilisateur avec ces systèmes soulève plusieurs questions (Bown et autres, 1979). Comment, par exemple, le public demandera-t-il l'information des diverses bases de données externes qu'il aura à sa disposition? En effet, il existe actuellement un certain nombre de langages, tous différents les uns des autres, ayant chacun son vocabulaire et sa syntaxe propres.

Il semble peu souhaitable, du moins à première vue, d'obliger l'utilisateur d'un système vidéotex à apprendre et retenir un langage d'interrogation distinct pour chaque base de données externe qu'il voudrait consulter. Pour que les utilisateurs acceptent les services vidéotex de recherche de l'information, il ne devrait y avoir qu'une seule manière d'interroger et le public devrait pouvoir l'apprendre et l'utiliser facilement. Avec ce langage d'interrogation, les gens pourraient obtenir l'information de toute base de données externe reliée au réseau de communication.

Dans le présent rapport, nous définissons et décrivons plusieurs catégories de langages d'interrogation interactive. Nous établissons quelques critères d'interaction des usagers qui permettent d'évaluer ces catégories de langages d'interrogation. Nous faisons une brève étude de chaque catégorie en fonction des critères énumérés ci-dessus et, en dernier lieu, nous proposons une méthode particulière d'aborder un langage d'interrogation pour les bases de données externes.

2 PARAMÈTRES D'INTERACTION DE L'UTILISATEUR

L'interaction personne-ordinateur, a fait l'objet de nombreuses études récentes (Martin, 1973; Gilb et Weinberg, 1977; Guedj et autres, 1980; Schneiderman, 1980; Mehlmann, 1981). Cependant, beaucoup moins nombreuses sont celles qui portent sur les langages d'interrogation interactive (Reisner, 1981). En utilisant certains résultats des études qui ont porté respectivement sur l'interaction personne-ordinateur et sur les langages d'interrogation interactive, nous étudierons le processus d'interrogation de la base de données du point de vue de l'utilisateur. On peut diviser le processus en trois parties: la demande, la réponse et la dynamique.

La demande a trait à la formulation de l'interrogation du système par l'utilisateur. Ce dernier informe le système de ce qu'il attend de lui. Ce qui nous intéresse ici, ce sont les demandes d'information en mémoire. On utilise habituellement une forme de **langage** que l'utilisateur et le système acceptent tous deux. Le langage peut prendre plusieurs formes: visuel, oral, gestuel, etc.

La réponse vise à informer l'utilisateur du résultat de l'action. Ce résultat peut être la réponse à la demande ou une information sur le statut d'une demande. Les données de sortie devraient être présentées afin que l'utilisateur puisse facilement les comprendre et les assimiler. Encore là, il faut une forme de langage pour communiquer les données de sortie. La visualisation semble être un choix naturel pour le système vidéotex. Il peut s'agir de tableaux, de graphiques, d'images, etc. La présentation sonore est une autre solution possible.

Enfin, **la dynamique** a trait à la nature de l'interaction entre l'utilisateur et le système. Les processus de demande et de réponse

doivent se faire par un moyen qui permette à l'utilisateur et au système de communiquer l'un avec l'autre, c'est-à-dire que les langages doivent disposer de moyens de transport entre les deux **participants**. Ainsi, l'interaction peut se faire par clavier, oralement, par pointage, etc. Du point de vue de l'utilisateur, la qualité de l'interaction est un aspect très important du processus d'interrogation. Dans un système vidéotex, l'interaction devrait, selon nous, être intéressante et facile pour l'utilisateur.

Dans la plupart des langages d'interrogation, les trois parties mentionnées ci-dessus sont considérées en bloc. C'est là une erreur puisque le langage utilisé pour formuler les demandes n'a pas, pour l'utilisateur, les mêmes exigences qu'un langage utilisé pour donner les réponses. De plus, la dynamique d'un langage est indépendante des aspects statiques de la formulation des demandes et de la présentation des réponses. Par conséquent, on peut étudier les exigences relatives à la dynamique indépendamment des exigences de la demande et de la réponse.

On peut dès lors définir certaines caractéristiques pour chaque partie du processus d'interrogation. Ces caractéristiques décrivent les paramètres de l'interaction des utilisateurs. Les **valeurs** qu'affichent ces paramètres dans un langage d'interrogation donné déterminent si ce langage a, ou n'a pas, les caractéristiques **souhaitables** en ce qui a trait à l'interface personne-ordinateur. Cette section sera consacrée à l'identification et à la définition de plusieurs caractéristiques pour chaque partie du processus d'interrogation. De plus, nous indiquerons ce qui, selon nous et selon les études mentionnées ci-dessus, constitue une valeur **souhaitable** pour chaque caractéristique. La liste des valeurs **souhaitables** constituera alors les exigences d'un langage **souhaitable** pour l'interrogation interactive.

2.1 Demande

Nous identifions ici six caractéristiques de formulation des demandes qui sont importantes pour l'évaluation des langages d'interrogation interactive.

FRAPPES

Les frappes sont la quantification des données introduites par l'utilisateur pour que le système comprenne entièrement la demande de l'utilisateur. Si l'on utilise un clavier pour formuler la demande, la mesure se fera en comptant le nombre de touches frappées. Si la demande est formulée en pointant des articles, c'est le nombre d'articles pointés qui constituera la mesure. Il est souhaitable de **réduire au minimum** le nombre de frappes nécessaires à la formulation de la demande. Ainsi, la formulation sera rapide et facile et l'on évitera les possibilités d'erreurs, les mauvaises touches, par exemple.

COMMANDES

Les commandes sont l'ensemble des instructions dont dispose l'utilisateur pour signifier au système ce qu'il a à faire. Ainsi, l'utilisateur peut demander au système de **rechercher**, **d'imprimer** ou de **dessiner**. Le nombre des commandes doit être **restreint** pour qu'on puisse les mémoriser facilement. Les commandes doivent être **simples**, c'est-à-dire que la syntaxe d'une commande et le nombre de choses à spécifier pour utiliser la commande doivent être faciles à mémoriser et l'utilisateur doit pouvoir s'en servir sans hésiter.

FORMULATION

La formulation d'une demande concerne la difficulté plus ou moins grande que l'on a à spécifier une demande dont la syntaxe ou la sémantique sont incorrectes. Une demande dont la syntaxe est incorrecte ne suit pas la structure que reconnaît le système. Il s'agit souvent de mots comportant une faute d'orthographe ou encore d'une demande incomplète. Une demande incorrecte sur le plan de la sémantique en est une à laquelle le système ne peut donner de réponse parce qu'il n'a pas l'information requise; cela peut encore être une demande pour laquelle il y a eu une réponse qui n'est pas celle que l'utilisateur attendait, c'est-à-dire, que l'interprétation de la demande par le système diffère de celle de l'utilisateur. Il est souhaitable qu'un langage d'interrogation élimine la possibilité de demandes contenant des erreurs de syntaxe ou de sémantique ou, du moins, qu'il rende difficile la formulation de demandes contenant de telles erreurs.

SÉLECTIVITÉ

La sélectivité est la possibilité qu'a l'utilisateur de spécifier aussi précisément que possible les données qu'il désire retrouver. Si un langage d'interrogation comporte une grande sélectivité, l'utilisateur n'obtiendra que les données voulues. Si, par contre, la sélectivité en est faible, beaucoup des données recouvrées seront du **bruit**, c'est-à-dire qu'elles ne répondront pas à la demande. Il est donc souhaitable que le langage ait une **grande** sélectivité, pour que l'utilisateur ne soit pas distrait par des données parasites. De plus, si la facturation se fait en fonction des données recouvrées, une grande sélectivité peut contribuer à réduire les coûts d'utilisation du système.

UNIFORMITÉ

L'uniformité concerne l'indépendance du langage d'interrogation par rapport au genre de base de données à laquelle il donne accès et l'application pour laquelle le langage est utilisé. Pour des bases de données, qui peuvent être de divers types et structurées de façons différentes, la question de l'uniformité se pose ainsi: le langage d'interrogation peut-il être utilisé pour accéder à toutes ces bases de données? Pour les applications, le langage peut-il être utilisé pour tout genre d'application ou la structure de l'application est-elle spécifique? Ainsi, si l'on peut utiliser un langage d'interrogation pour avoir accès à de l'information sur la bourse, la météo, les nouvelles, l'épicerie, etc., on a un langage uniforme, du point de vue de l'application. Il est souhaitable que la structure d'un langage d'interrogation soit uniforme, et pour les bases de données et pour les applications. Dans la pratique, il est facile d'obtenir l'uniformité d'application, tandis qu'il est beaucoup plus difficile d'obtenir l'uniformité dans le cas des bases de données.

PERSONNALISATION

Quand on parle de personnaliser un langage d'interrogation, cela signifie en adapter la forme à celle d'une application particulière. Cela a pour but de faciliter la formulation des demandes en fonction de l'application, puisque la forme du langage d'interrogation aide l'utilisateur à mémoriser le langage. Ainsi, certaines applications utilisent, pour des opérations données, des mots spécifiques qui pourraient servir de mots-clés dans un langage d'interrogation. De cette façon, la sémantique d'une demande est acheminée jusqu'à l'utilisateur plus facilement et avec plus de précision. Il est souhaitable qu'un langage d'interrogation fasse appel à autant de concepts d'application que possible pour faciliter l'interaction de l'utilisateur avec le système.

2.2 Réponse

Nous identifions ici cinq caractéristiques qui servent à représenter les réponses; ces caractéristiques sont importantes pour l'évaluation des langages d'interrogation interactive.

PRÉSENTATION

La présentation est la forme dans laquelle la réponse est donnée à l'utilisateur. Le facteur le plus important est la complexité de la présentation. Une façon de mesurer cette complexité consiste à mesurer le temps que met l'utilisateur à saisir le contenu de la réponse. Si la réponse est donnée sous la forme d'un dialogue que l'utilisateur doit lire sur un écran de visualisation, il faudra peut-être un certain temps pour comprendre l'information donnée en réponse. Par contre, les graphiques ou les tableaux peuvent ordinairement être interprétés beaucoup plus rapidement. De plus, le même espace permet de présenter beaucoup plus d'information. Il est souhaitable que le langage d'interrogation affiche la réponse à une demande sous une forme que l'utilisateur comprendra et suivra sans difficulté.

PRÉSENTATION SUR SUPPORTS MULTIPLES

Les gens communiquent de plusieurs façons. Par conséquent, s'il est souhaitable que l'utilisateur puisse choisir la forme que prendra une réponse, il est également souhaitable qu'il puisse choisir le support qui servira à lui présenter cette réponse. Ainsi, la réponse pourrait être imprimée sur papier, affichée sous forme de texte, de graphique, ou encore présentée sous forme d'animation sur un écran; elle pourrait même être vocale. De cette façon, les besoins et l'humeur de l'utilisateur détermineraient la formulation de la réponse. Il pourrait vouloir parcourir la réponse à tête reposée, y jeter un coup d'oeil rapide ou encore l'écouter tout en faisant autre chose. Il peut aussi

être important d'offrir le choix entre divers appareils de présentation et divers supports de communication, de façon à permettre aux personnes handicapées d'accéder au système.

PERSONNALISATION

Les langages d'interrogation ne tiennent ordinairement pas compte des personnes auxquelles sont destinées les réponses. Il y a habituellement une seule manière de présenter les réponses, peu importe l'application, le genre d'appareil d'affichage, le moment de la journée et le type d'utilisateur. La personnalisation des réponses, tout comme celle des demandes, signifie que la forme de la réponse est déterminée par les facteurs qui entourent l'interaction. Ainsi, l'affichage de l'information météo se fera mieux graphiquement que par des textes. Les enfants peuvent vouloir obtenir les réponses à leurs questions sous forme de dessins animés qu'ils comprendront mieux que les textes ou les graphiques, plus difficiles à interpréter. Les utilisateurs peuvent choisir des formes qui varieront selon le moment de la journée: en effet, ils préféreront peut-être de courts résumés verbaux tard la nuit tandis qu'ils voudront des textes plus détaillés en début de soirée. Il est souhaitable qu'un langage d'interrogation donne toutes les possibilités de personnaliser les réponses données à l'utilisateur selon les facteurs énumérés ci-dessus.

COMMANDE DYNAMIQUE

L'utilisateur devrait pouvoir régler le rythme auquel la réponse lui est présentée. Cela signifie que, si la réponse est donnée à l'utilisateur sous une certaine forme et en utilisant un certain support, le système doit donner à l'utilisateur un moyen de commander efficacement la présentation. L'utilisateur devrait pouvoir commander la vitesse à laquelle le texte est affiché à l'écran. Il devrait aussi pouvoir commander dynamiquement la vitesse, c'est-à-dire pouvoir revenir

en arrière ou encore sauter des parties de texte. Il devrait aussi pouvoir changer la forme de la réponse, passer du texte à la parole, par exemple, sans devoir reformuler sa demande à nouveau. De plus, ces changements ne devraient pas présenter trop de difficultés.

RÉUTILISATION POSSIBLE

Il arrive souvent que la réponse à une demande ne serve pas immédiatement mais qu'elle soit utile plus tard ou puisse servir comme donnée d'entrée pour une autre demande. Il est donc souhaitable que le langage d'interrogation permette de conserver la réponse à une demande pour la traiter par la suite ou pour la réutiliser. De cette façon, il est possible de ramener les interrogations complexes à des fragments logiques plus simples et de réduire ainsi la quantité totale des interactions entre l'utilisateur et le système. Ce mode d'interaction peut se rapprocher davantage de la manière dont les gens s'acquittent de tâches complexes.

2.3 Dynamique

Nous identifions ici cinq caractéristiques d'interaction entre l'utilisateur et le système qui sont importantes pour l'évaluation des langages d'interrogation interactive.

INTENSITÉ

L'intensité de l'interaction est la mesure de la vitesse à laquelle l'utilisateur et le système communiquent entre eux. En général, les gens peuvent absorber plus d'information sous forme d'images que sous forme de texte. L'utilisation d'un clavier pour faire une demande exige beaucoup moins d'intensité que la voix. L'intensité dépend des moyens techniques disponibles pour l'interaction et de la manière dont les demandes et les réponses sont énoncées. Il est souhaitable

d'optimiser l'intensité des communications entre l'utilisateur et le système par l'intermédiaire du langage d'interrogation.

LUDICITÉ

Il est important que l'interaction entre le système et l'utilisateur soit intéressante pour ce dernier. La nature de l'interaction doit donc représenter un défi et le degré de difficulté de l'interaction doit être lié à l'avantage que l'on tire de l'information obtenue. On appelle ce paramètre la ludicité. L'utilisateur du système vidéotex n'a aucune motivation extérieure, par exemple financière, qui puisse le pousser à une interaction avec le système. La motivation est purement intérieure, c'est-à-dire que l'utilisateur veut apprendre quelque chose ou encore se détendre. Il doit sentir que le temps qu'il investit vaut les résultats obtenus, ou du moins que le temps consacré à l'interrogation a été une source de joie et non de frustration. Par conséquent, si la formulation des demandes est intéressante et synonyme de défi, le sentiment d'ennui qu'entraîne l'interaction peut être atténué par sa valeur éducative ou récréative.

PROTOCOLE

Le protocole est la manière spécifique dont l'utilisateur dialogue avec le système. Ainsi, l'utilisateur peut toujours formuler une demande de la même façon en se servant du clavier. Un langage d'interrogation devrait offrir plusieurs protocoles d'interaction entre l'utilisateur et le système. De cette façon, l'utilisateur peut à tout moment choisir le protocole qu'il désire. Il peut être souhaitable de disposer de différents protocoles selon la difficulté des demandes. De cette façon, les demandes simples sont faciles à formuler, un seul bouton à presser par exemple, tandis que les demandes plus complexes exigent plus de travail.

SOUPLESSE

L'idéal serait que le système réponde toujours immédiatement à la demande de l'utilisateur. Cependant, ce n'est pas toujours techniquement possible. À défaut de cette situation idéale, la réponse du système devrait être adaptée à la nature de l'interaction. Par conséquent, si une demande est courte ou très facile à formuler, l'utilisateur peut s'attendre à ce que le système donne rapidement une réponse. Inversement, si la demande est longue ou très difficile à formuler, on peut raisonnablement s'attendre à ce que la réponse tarde à venir. Il est souhaitable que la difficulté à formuler la demande entraîne de la difficulté à la traiter. De cette manière, l'utilisateur n'est pas ennuyé d'avoir à attendre pour obtenir la réponse à une question qu'il trouve très facile à formuler.

MAÎTRISE

Les gens aiment avoir la sensation de contrôler la situation, par eux-mêmes ou par personne interposée. Ils n'aiment pas se sentir menés, et surtout pas par une machine. Quand il dialogue avec un système informatisé, l'utilisateur devrait toujours avoir l'impression qu'il domine la situation et que le système vient l'appuyer. Même si, finalement, c'est le système qui a la maîtrise, il doit susciter chez l'utilisateur le sentiment que c'est plutôt lui qui dirige. La souplesse et la serviabilité du système sont des facteurs déterminants qui contribuent à donner cette sensation. Le degré d'interaction est aussi important. Par exemple, on a davantage l'impression de contrôler la situation quand on est au volant d'une automobile plutôt que dans un autobus en tant que passager passif. Par conséquent, à la suite d'une demande, la réponse du système contribue à donner l'impression que l'on domine la situation. Cette impression peut être le facteur déterminant qui incitera l'utilisateur à accepter le système.

2.4 Résumé

Nous avons exposé à grands traits plusieurs des caractéristiques souhaitables d'un langage d'interrogation. Il arrive quelquefois que les exigences relatives aux caractéristiques souhaitables s'harmonisent bien. Ainsi, l'intensité de l'interaction peut être optimisée en diminuant le nombre de frappes par unité de temps. Dans d'autres cas, ces exigences s'opposent les unes aux autres. Ainsi, plusieurs protocoles dans un langage d'interrogation peuvent nuire à l'uniformité, puisque des protocoles différents peuvent faire appel à des commandes différentes. Il s'agit donc, au moment de concevoir un langage d'interrogation, de définir et d'évaluer les compromis que l'on doit faire entre les diverses exigences pour en arriver à un choix acceptable de paramètres d'interaction pour l'utilisateur. Plutôt que de choisir une valeur, il peut être souhaitable de donner plusieurs versions d'un langage d'interrogation dans lequel on met l'accent sur différentes exigences. L'évaluation des facteurs humains peut aider à décider des meilleures approches et à choisir la valeur des paramètres.

Pour montrer comment on évalue l'interaction de l'utilisateur par rapport à ces exigences, on peut étudier une demande portant sur les prix à la bourse ou sur les rapports météorologiques. Une façon de fournir ce service consiste à avoir un rapport complet sur la bourse ou la météo disponible sur demande. La personne presse un bouton du logiciel identifié **bourse** ou **météo** et le système lui fournit, sous la forme d'un texte suivi, un rapport normalisé sur la bourse ou la météo. Jetons maintenant un coup d'oeil sur les paramètres de l'interaction de l'utilisateur pour évaluer dans quelle mesure ce service est souhaitable.

Pour ce qui est de la formulation de la demande, le nombre de frappes est minimal, la commande est simple, les mauvaises demandes ne peuvent pas être formulées et ce genre de commande peut convenir à

beaucoup d'applications (en utilisant des boutons différents) et peut servir à de nombreuses applications. Cependant, la commande n'a absolument aucune sélectivité. L'utilisateur aura toujours un rapport complet sur la bourse ou la météo. Il ne peut pas isoler un secteur d'affaires, une compagnie ou un moment choisi, pour ce qui est de la bourse, ni une région géographique ou un moment particulier pour ce qui est du rapport météorologique. Il obtient tout ce qu'il y a en mémoire et il a la tâche d'en extraire les données de qui l'intéressent.

Quant à la réponse, la présentation est assez simple et il lui suffit de la faire se dérouler à la vitesse qui lui convient et d'habitude il peut conserver la réponse pour consultation ultérieure. Cependant, l'utilisateur n'a pas souvent le choix du moyen ni du format de présentation de l'information, et ne peut pas non plus adapter la réponse aux possibilités particulières de son terminal, celle de produire des graphiques, par exemple. Il obtient la réponse sous forme de texte parce que cela convient à tous les appareils de sortie. Cependant, la météo peut prendre différentes formes selon que l'utilisateur est un enfant, qui préfère des dessins animés, ou bien un homme d'affaires qui aura besoin d'un rapport précis et concis.

Passons maintenant à la dynamique de l'interaction. Il n'y en a pratiquement pas. La demande est formulée en pressant un bouton. L'intensité est alors très grande du fait de la rapidité avec laquelle l'utilisateur communique la demande au système, mais, par la suite, l'interaction est terminée pour ce qui est de l'utilisateur. Il ne peut intervenir sur la réponse: il est complètement passif. Ce sont là des conditions stériles et ennuyantes pour choisir de l'information.

3 ÉVALUATION DU LANGAGE D'INTERROGATION

Dans la présente section, nous répartissons les langages d'interrogation en sept catégories. Pour chacune d'elles, nous décrirons brièvement la forme des langages. Par la suite, nous parlerons du genre de langage par rapport aux paramètres d'interaction de l'utilisateur décrits à la section 2.

3.1 Langages à mots-clés

Les langages d'interrogation à mots-clés permettent à l'utilisateur de se servir d'une forme restreinte de langue courante pour dialoguer avec le système. Il existe de nombreux exemples de ces genres de langages dont le SQL (Denny, 1977), le ILL (Lacroix et Pirotte, 1977 a, b), et le NUL (Deheneffe et Hennebert, 1976). Les demandes sont formulées en utilisant une grammaire restreinte semblable à celle de l'anglais dont des mots précis (mots-clés) signalent le début d'éléments spécifiques d'une demande. Pour faciliter le traitement de ces langages, la syntaxe des demandes est très rigide. Les éléments doivent être utilisés dans un ordre et un format fixes. Les réponses aux demandes sont présentées à l'utilisateur sous forme de tableaux.

Pour formuler une demande, l'utilisateur doit faire appel à un bon nombre de frappes. Cependant, une fois que la demande a été formulée, elle est très facile à lire et à interpréter puisque la structure de ces langages est étroitement liée à la structure de la phrase anglaise. D'habitude, il n'y a que très peu de commandes, mais elles sont assez complexes et offrent de nombreuses possibilités quant aux exigences précises de l'utilisateur. Par conséquent, il faut un certain temps pour maîtriser le langage en entier, quoique l'on puisse en apprendre des sous-ensembles assez rapidement (Reisner et autres, 1975; Reisner, 1977). Comme chaque commande est complexe et le nombre de frappes élevé, le risque de commettre des erreurs de syntaxe dans les

demandes est élevé. Cependant, ces erreurs sont immédiatement repérées par le système. Il est possible de formuler des demandes sans signification, du point de vue de la sémantique, mais cela devient peu probable si la demande ne contient pas d'erreurs de syntaxe et que la base de données est bien conçue. Les langages d'interrogation à base de mots-clés ont une très grande sélectivité. En effet, l'utilisateur peut spécifier très précisément les données qu'il veut obtenir. Ces langages ne tiennent pas compte de la nature de la base de données, quant à son application, bien qu'ils supposent une certaine structure globale des données, qui peut être relationnelle, par exemple. Donc, ces langages ne sont pas liés à l'application; cependant, les bases de données doivent être d'un genre bien précis. Les mots-clés d'un langage peuvent être personnalisés en fonction d'applications particulières. Les autres aspects de ces langages sont rigides.

Les réponses aux questions sont habituellement présentées sous forme de tableaux. L'utilisateur n'a aucune possibilité de changer la forme de la sortie. Bien que les tableaux soient assez faciles à saisir pour certains genres de sortie, ils peuvent être très difficiles à interpréter s'il existe des relations complexes entre les tableaux. Les données de sortie sont normalement présentées sur écran ou sur imprimante seulement. Ces langages ne permettent pas de personnaliser la sortie pour l'utilisateur. Dans la plupart des cas, les données de sortie sont présentées avec des titres de tableaux et de colonne que l'utilisateur peut modifier à sa guise. L'utilisateur n'a aucune commande dynamique sur la sortie à l'exception de celle que fournit la nature même de l'appareil d'affichage, comme le défilement, par exemple. La plupart des langages à base de mots-clés permettent à l'utilisateur de conserver les résultats d'une demande pour les utiliser par la suite.

La dynamique de l'interaction est très minime dans le cas des langages à base de mots-clé. L'intensité de l'interaction est très

faible. Les demandes doivent être introduites en utilisant un clavier et les réponses, lues sur un écran ou une imprimante. L'interaction ne comporte pratiquement aucune ludicité: tout est décidé d'avance. Il n'existe qu'un protocole pour communiquer avec le système: introduire les demandes avec un clavier dans la forme prescrite par le langage. Dans quelques langages d'interrogation à base de mots-clés, les demandes qui sont difficiles à traiter, comme les sommes logiques dans les systèmes rationnels, sont très simples à préciser. Par conséquent, la souplesse du système ne correspond pas à la complexité de la formulation de la demande. Comme les demandes ne peuvent être formulées que de façon restreinte, le système ne peut pas donner beaucoup de rétroaction avant que l'ensemble de la demande n'ait été précisé. Dans de telles conditions, l'utilisateur n'a pas vraiment l'impression de contrôler le système.

3.2 Langages par exemples

Dans les langages d'interrogation **par exemples** l'utilisateur élabore sa demande en donnant au système un exemple de réponses à sa demande. L'exemple est construit selon le format utilisé par le système pour fournir les réponses. Ainsi, dans le rapport **Interrogation par exemples** (Zloof, 1975a; Zloof, 1977, 1980) des modèles de tableaux sont affichés sur un écran et l'utilisateur remplit le tableau selon les colonnes qu'il désire voir dans sa réponse et selon les valeurs qu'il désire dans chaque colonne. Pour le TLA, des modèles de formules sont remplis pour préciser une interrogation (Hogg, 1981; Hogg et autres, 1981; Nierstrasz, 1981). D'autres chercheurs ont proposé des langages semblables (Hammer et autres, 1977; Luo et Yao, 1981).

Dans ces genres de langage d'interrogation, bien que l'utilisateur ait à faire une certaine quantité d'entrées, le nombre de frappes est relativement limité. Le système fournit beaucoup de données d'entrée comme les noms d'attributs qui seraient nécessaires pour les langages à

mots-clés. L'utilisateur n'a qu'à indiquer (normalement par une seule frappe) les valeurs d'attribut qu'il désire dans la réponse et toute condition qui détermine le choix des valeurs. Les commandes qui indiquent l'opération désirée sont généralement limitées à des codes à un seul caractère. Les commandes de ces langages sont simples et peu nombreuses. Cependant, elles ne se prêtent pas bien à la formulation de demandes complexes qui s'étendent sur plusieurs modèles. On sacrifie quelque peu la lisibilité au profit de la simplicité de l'expression. Il est très difficile de formuler des demandes contenant des erreurs de syntaxe, étant donné la rigidité de la spécification. C'est pourquoi il est difficile de formuler des demandes comportant des erreurs de sémantique. En effet, l'utilisateur est toujours au courant de l'information que possède le système puisqu'elle lui est présentée dans les modèles et seules les demandes concernant cette information peuvent être formulées. La sélectivité de ces langages d'interrogation est comparable à celle des langages à mots-clés. La manière dont les demandes sont précisées ne dépend pas de l'application: seuls les modèles changent, et non la manière dont on les remplit. En autant que les bases de données auxquelles on a accès peuvent être représentées sous la forme des modèles utilisés par le langage, on peut dire que le langage n'est pas lié lui non plus à la base de données. La personnalisation de ces langages en fonction d'une application ne peut concerner que les noms d'attributs à l'intérieur d'une application. Comme la forme du langage est rigide, il est difficile de tirer profit de toute caractéristique spéciale d'une application.

Les réponses à des demandes se présentent à l'utilisateur sous la forme de modèles qui servent à formuler les demandes. Par conséquent, il y a correspondance directe entre les demandes et les réponses. De plus, comme les demandes prennent habituellement la forme de tableaux ou de formulaires, on peut les suivre et les comprendre assez facilement. Cependant, cette caractéristique restreint aussi la réponse du fait

qu'elle est présentée sur un écran ou sur une imprimante. Il est aussi difficile de personnaliser les réponses que les demandes. En effet, la forme des réponses est dictée par celle des demandes. Étant donné la nature des réponses, la seule commande dynamique possible est le réglage du défilement sur l'écran. Cependant, ces langages permettent ordinairement à l'utilisateur de conserver les résultats des demandes pour utilisation ultérieure.

Pour ce qui est de la dynamique, nous nous trouvons dans des conditions de demande/réponse pour lesquelles les deux processus sont disjoints. L'intensité est assez grande pour ce qui est du temps consacré à préciser la demande et elle est liée à la complexité de la demande. Ainsi, une demande simple peut être précisée rapidement, tandis qu'une demande plus complexe exige plus de travail. L'intensité de sortie est limitée par la vitesse à laquelle l'utilisateur peut lire ce que lui présente l'écran. Ces genres de langages ont une ludicité un peu plus grande que les langages à mots-clés. Le travail de spécification de la demande est moins ennuyeux et il y a même des systèmes qui guident l'utilisateur quand il remplit les tableaux ou les formules, en ce sens que le curseur passe directement à l'attribut suivant sans que l'utilisateur n'ait à provoquer son mouvement. Par contre, il doit toujours spécifier sa demande en remplissant des modèles. En ce qui a trait à la souplesse, nous devons rappeler que certaines demandes, en particulier celles pour lesquelles il faut passer d'un modèle à un autre, présentent quelques difficultés de formulation. Les demandes difficiles à formuler sont aussi difficiles à traiter. Étant donné le contexte demande/réponse, l'utilisateur n'a pas un contrôle très précis du système. Bien sûr, le système peut donner quelques directives mais sa contribution est restreinte.

3.3 Langage naturel

Les langages d'interrogation basés sur le langage naturel permettent à l'utilisateur anglophone de préciser sa demande dans sa langue maternelle. Le but visé n'est pas de restreindre le vocabulaire de quelque façon que ce soit, bien que, dans la pratique, le système ne puisse traiter qu'une partie d'une langue. Il y a beaucoup d'exemples de ces systèmes qui ont été mis au point par des chercheurs oeuvrant dans le domaine de l'intelligence artificielle (Woods, 1973; Mylopoulos et autres, 1975, 1976; Harris, 1977; Hendrix et autres, 1978). Le système RENDEZ-VOUS mis au point par le laboratoire de recherche IBM à San Jose est peut-être l'exemple le plus connu d'un système qui utilise la technique de la base de données (Codd et autres, 1978).

L'utilisateur formule ses demandes en dialoguant avec le système. Cela signifie qu'il doit utiliser un très grand nombre de frappes pour préciser sa demande. Il n'y a pas de commandes fixes comme telles. Comme c'est le cas pour les phrases énoncées en anglais, les demandes simples sont assez faciles à préciser, et les demandes plus complexes, plus difficiles. Comme l'utilisateur a pleine liberté d'utiliser toutes les ressources de la langue anglaise, il lui est possible de formuler sa demande de plusieurs façons. Cela implique qu'il risque de faire plusieurs genres d'erreurs augmentant ainsi les risques d'une mauvaise interprétation (Schneiderman, 1978). Pour ce qui est de la syntaxe, ces systèmes possèdent dans bien des cas un mécanisme qui permet de corriger les fautes d'orthographe. Ces langages peuvent être très sélectifs, mais cela dépend de la précision de l'utilisateur lorsqu'il exprime sa demande. D'ailleurs, la spécification de la demande peut poser des difficultés puisque le langage ne donne aucune directive à cet égard. Le langage naturel est le plus uniforme que l'on puisse utiliser pour préciser une demande. En effet, il est uniforme pour les applications comme pour les bases de données. De plus, le langage peut être personnalisé par l'utilisateur en fonction de l'application; pour cela, il n'a qu'à faire appel à ses connaissances sur l'application. Cela

suppose que le système possède quelques connaissances sur l'application lui permettant de comprendre le vocabulaire de l'utilisateur.

Les réponses aux demandes peuvent être présentées sous forme de phrases anglaises pour des faits uniques. Pour des faits multiples, il pourrait se présenter des difficultés. Les réponses sont souvent présentées à l'écran sous forme de phrases. Cependant, il est possible d'adapter les phrases à un appareil de sortie vocale. La forme de sortie, tout comme les phrases anglaises, se prête à la personnalisation. Il est possible d'utiliser des mots qui appartiennent à une application ou à un type d'utilisateurs. Parce que l'interaction se fait sous forme de dialogue, l'utilisateur peut exercer une commande dynamique sur la réponse à sa question. L'utilisateur peut communiquer avec le système jusqu'à ce que la réponse contienne l'information qui l'intéresse. Il est difficile de conserver ou de réutiliser la réponse à moins que le système soit capable d'accepter tout nouveau fait qui lui est fourni sous forme de phrases en langage naturel.

La dynamique de l'interaction consiste en un dialogue entre l'utilisateur et le système. Pour les entrées faites par l'intermédiaire du clavier, l'intensité est très faible puisqu'il faut composer des phrases entières en utilisant le clavier. Si la sortie se fait sous la forme de phrases écrites à la machine, l'intensité est aussi très faible. Comme l'utilisateur dialogue avec le système, ce dernier peut introduire une certaine ludicité dans l'interaction, selon la façon employée pour répondre aux demandes de l'utilisateur. Ainsi, il peut demander à l'utilisateur de clarifier une partie de sa demande qu'il ne comprend pas. Il peut reformuler en d'autres termes la demande de l'utilisateur. Cela accorde aussi à l'utilisateur la possibilité de communiquer avec le système de plusieurs façons, par exemple énoncer des phrases, répondre aux questions du système ou choisir entre des possibilités que lui propose le système. Comme les demandes sont formulées par éléments, l'utilisateur a toujours l'illusion qu'il

progresses vers la réponse qu'il cherche. Ce processus permet aussi au système de traiter des demandes complexes en éléments séparés, ce qui a pour effet de réduire les délais entre la formulation de la demande et la réponse. Le processus de formulation de la demande par éléments donne à l'utilisateur l'impression qu'il exerce un contrôle sur le système qu'il guide vers la réponse.

3.4 Menu

Les langages d'interrogation basés sur le menu permettent à l'utilisateur de préciser sa demande en pointant l'action ou les données désirées. À tout moment, l'utilisateur peut choisir dans le menu des données ou des opérations possibles. Le choix d'une opération ou d'un objet peut amener la présentation d'autres menus à l'utilisateur. De cette façon, les demandes sont précisées par étapes selon une structure hiérarchique. On a souvent utilisé ce genre de systèmes et pour beaucoup d'applications. Le système Officetalk mis au point par le centre de recherche de Xerox à Palo Alto, est un exemple d'un système qui interroge les bases de données à l'aide de menus. (Ellis et Nutt, 1980).

Les demandes sont précisées en pointant les actions désirées. Par conséquent, il y a très peu de frappes à utiliser. Les commandes sont simples et elles sont représentées soit par des mots-clé anglais, soit par des icônes qui représentent les actions ou les objets désirés. Ainsi, pour mettre en mémoire un document qui se trouve sur l'écran, on n'a qu'à pointer le document et l'image d'un classeur. Étant donné la structure hiérarchique des menus, les commandes donnent un résultat satisfaisant quand les demandes sont simples et ont une structure hiérarchique, mais elles posent des difficultés dans le cas de demandes plus complexes. La nature très hiérarchisée de l'interaction rend improbable la formulation de demandes incorrectes. C'est le système qui dirige l'interaction à toutes les étapes. La sélectivité est très bonne

pour les demandes qui étaient prévues, c'est-à-dire qui suivent la structure du menu, mais elle est mauvaise pour les autres demandes. On peut utiliser la même technique de choix au menu, peu importe l'application et le genre de base de données. Il est possible de personnaliser les menus selon des applications spécifiques en changeant les mots-clés ou les icônes utilisés.

Les réponses aux demandes sont présentées à l'utilisateur sous forme d'objets stockés dans le système. Dans la plupart des systèmes qui font appel à la technique du menu, il y a des formules qui contiennent un texte structuré ou un tableau. Par conséquent, les réponses présentées de cette manière sont plutôt faciles à suivre. La sortie est disponible seulement sur un écran ou une imprimante et l'utilisateur doit y lire la réponse. Si le système permet de définir de nouveaux objets "au vol", comme c'est le cas avec l'Officetalk, on peut alors personnaliser la sortie pour l'utilisateur. Cependant, cette personnalisation ne porte que sur les genres d'objets conservés par le système, comme des formules, par exemple. L'utilisateur peut commander dynamiquement la présentation de la sortie s'il lui est possible d'utiliser les opérations de menu pendant la présentation de la sortie. Par conséquent, si la réponse peut prendre plusieurs formes, il peut alors les regarder à tour de rôle et les soumettre à une action avant de passer à la suivante. Il se peut que des dispositifs puissent conserver les données de la réponse et les réutiliser par la suite.

La dynamique de l'interaction, pour ces genres de langages, est quelque peu meilleure que pour ceux mentionnés ci-dessus. L'intensité de l'entrée est relativement bonne une fois que l'on a maîtrisé le mécanisme des opérations de pointage. Cette intensité peut être améliorée si l'on utilise des écrans qui permettent l'utilisation de plusieurs fenêtres. L'intensité de sortie est un peu restreinte parce

que l'utilisateur doit lire la réponse. L'interaction par pointage présente des possibilités intéressantes de ludicité. Elle est certainement moins ennuyante que l'utilisation du clavier, et l'emploi d'icônes pour représenter les actions peut contribuer à maintenir l'intérêt de l'utilisateur. Dans le cas du système Officetalk, l'utilisateur peut communiquer avec le système de diverses manières, par exemple il peut passer à un éditeur pour mettre en forme l'entrée et la sortie, ou il peut passer en mode de dessin pour faire des dessins à main levée. Étant donné que l'opération de pointage de la part de l'utilisateur a une grande intensité, l'utilisateur peut s'attendre à ce que toutes les opérations soient d'une même complexité. Cependant, ce n'est pas normalement le cas. Par conséquent, la vitesse et le temps de réponse peuvent ne pas être uniformes, ce qui peut ennuyer l'utilisateur. La nature de l'interaction et sa grande intensité donnent à l'utilisateur l'impression qu'il a la maîtrise du système, qu'il le dirige toujours. Cependant, cela peut ne pas être le cas pour des demandes qui ne suivent pas la structure des menus.

3.5 Graphiques

Les langages d'interrogation à base de graphiques permettent de formuler les demandes surtout sous forme de graphiques, mais il faut habituellement y ajouter quelques données à l'aide du clavier. Chaque genre d'objet de la base de données se voit attribuer une forme géométrique précise. L'utilisateur manipule alors ces formes pour spécifier ses demandes. On utilise des conventions spéciales pour préciser le choix de données spécifiques ou de relations entre les données. Le langage graphique LSL (Lipson et Lapczak, 1976) et le CUPID du système INGRES (McDonald et Stonebraker, 1975) constituent deux exemples de ce genre de langages.

Bien qu'il ne faille qu'un nombre minimal de frappes pour formuler une demande, l'utilisateur doit fournir une bonne quantité de travail

pour construire les diagrammes qui représenteront sa demande. Les objets géométriques doivent être manipulés sur l'écran et les options de traitement spécifiées pour certains d'entre eux. Une fois que la forme de la demande a été spécifiée par un diagramme, il est assez facile de demander l'action nécessaire. Par conséquent, les commandes sont simples; la difficulté réside dans la construction des diagrammes. Il est difficile de formuler des demandes contenant des erreurs de syntaxe puisque c'est le système qui dirige la manière dont les objets sont rassemblés et raccordés les uns aux autres. De même, il est difficile de faire des erreurs de sémantique, quoique cela soit toujours possible. Les langages d'interrogation à base de graphiques peuvent avoir le même degré de sélectivité que les langages à mots-clés puisqu'ils utilisent le même genre de constructions: celles-ci sont tout simplement spécifiées d'une autre manière. Ces langages peuvent servir pour toutes les applications qui utilisent un même genre de base de données, puisque les bases de données utilisent les mêmes genres d'objets pour représenter les données. Pour ce qui est des bases de données de types différents, ces genres de langages peuvent ne pas s'appliquer à moins que l'on spécifie des conventions qui concernent l'interprétation des objets géométriques dans différents contextes. La personnalisation en fonction des applications peut se faire pour les noms d'objets de données employés et les mots-clés utilisés dans le langage.

Les réponses aux demandes sont habituellement présentées sous forme de tableaux, tout comme pour les langages d'interrogation à mots-clés. Par conséquent, les réponses ont à peu près les mêmes caractéristiques en ce qui a trait à la présentation, aux supports multiples, à la personnalisation, à la commande dynamique et à la possibilité de réutilisation.

Dans le cas des langages d'interrogation par graphiques, la dynamique d'interaction n'est pas particulièrement bonne. Les images ont une grande intensité comme moyen de communication, mais on n'obtient pas une très grande intensité quand on les utilise pour formuler des demandes. Il faut beaucoup de temps et de travail pour formuler des demandes par diagrammes. Il peut être plus facile de spécifier la demande par d'autres mécanismes et d'utiliser les diagrammes pour afficher les spécifications, c'est-à-dire que c'est le système qui construit les diagrammes et non l'utilisateur. Les diagrammes sembleraient comporter un haut niveau de ludicité. Il est amusant de construire des diagrammes, mais seulement quand c'est facile de le faire. Ce n'est pas le cas actuellement, pour ces langages, particulièrement dans le cas des demandes complexes. De plus, les utilisateurs inexpérimentés peuvent ne pas savoir exactement comment formuler une demande sous forme de diagrammes. La formulation de demandes, même simples, par des graphiques, peut exiger beaucoup de temps. Par conséquent, l'utilisateur ne peut pas, d'après la formulation de la demande, juger de la complexité de celle-ci. De même, en ce qui a trait à la perception, le processus de spécification de la demande est lent et apparaîtra ainsi à l'utilisateur, étant donné l'interaction restreinte avec le système. On peut douter que l'utilisateur ait l'impression d'avoir la maîtrise de l'interaction.

3.6 Supports multiples

Les langages d'interrogation à supports multiples, comme leur nom le laisse entendre, permettent de communiquer avec le système de plusieurs manières. La formulation des demandes et la présentation des réponses se font sous forme de textes, de données mises en forme par des graphiques, de la voix, de la couleur, de l'animation, etc. Des icônes représentent les catégories de données qui se trouvent dans la base de données. Cette répartition en catégories peut être spécifiée selon

plusieurs niveaux d'abstraction. L'utilisateur peut parcourir les icônes à un niveau particulier et en choisir un en vue d'un traitement ultérieur. En choisissant un icône, l'utilisateur obtient un gros plan des données correspondantes, et soit d'autres icônes qui précisent la catégorie des données, soit les données elles-mêmes. Les icônes peuvent représenter des données textuelles, mises en forme, visuelles ou autres. Les travaux les plus avancés dans ce genre sont ceux du système de gestion des données spatiales qui a été mis au point par le MIT, et le CCA à Boston (Herot, 1980).

Il est possible de formuler les demandes de bien des manières; on peut pointer des icônes, entrer des énoncés par clavier, choisir des menus, etc. L'utilisateur peut choisir les mêmes données de différentes façons, selon son humeur ou ses connaissances de la base de données. Les commandes peuvent être aussi simples que le pointage ou aussi complexes que la spécification du langage d'interrogation à mots-clés. Étant donné la souplesse du processus de formulation des demandes, il est possible de faire fausse route dans la recherche des données. Cela est particulièrement vrai quand l'utilisateur ne fait que parcourir ce qu'il a sous les yeux. Le degré de sélectivité possible dépend en grande partie de la structure de la base de données. Ainsi, s'il y a beaucoup de niveaux d'abstraction entre le plus haut niveau et les données elles-mêmes, le processus de **rapprochement** peut alors être passablement sélectif; sinon, il se peut que l'on choisisse beaucoup de données non pertinentes. Bien sûr, l'utilisateur a toujours la possibilité de choisir un langage d'interrogation plus sélectif. Ces techniques peuvent être utilisées pour toutes les applications et tous les genres de bases de données. Cependant, la technique utilisée, comme le pointage par exemple, peut être plus appropriée que d'autres dans certaines situations. La personnalisation en fonction des applications se fait selon le genre de données stockées pour une application et selon la manière dont elles sont stockées, à chaque niveau d'abstraction d'une base de données.

Les réponses aux demandes dépendent du niveau d'abstraction auquel on se trouve ainsi que de la nature de la base de données. Ces réponses peuvent se présenter sous la forme de textes, d'images, de graphiques ou de toute autre manière, selon ce qui convient à la représentation des données. Cela permet de stocker les données sous une forme qui convienne à l'utilisateur à qui on les destine. Celui-ci peut diriger le système de façon à ce qu'il affiche la réponse d'une certaine manière ou moyennant un certain support. L'utilisateur a la maîtrise entière sur la sortie, sur son débit, sur la forme et sur l'endroit de cette sortie. Bien que les données de sortie puissent être conservées, il n'est pas toujours possible de les réutiliser comme entrée pour une autre demande, si par exemple, elles sont sous forme de graphiques ou sous forme verbale.

La dynamique de l'interaction, est un excellent système. L'intensité de l'entrée et celle de la sortie sont grandes et l'utilisateur peut choisir la forme qui correspond à ses goûts. L'interaction a un degré élevé de ludicité. L'utilisateur peut **survoler** les données et obtenir très facilement un **gros plan** d'une partie en particulier. L'interaction avec le système peut se faire de plusieurs manières et être modifiée au moment de la formulation d'une demande. L'utilisateur peut répondre très rapidement à une question posée par le système. Cependant, il se peut que le système mette beaucoup de temps à répondre à une demande de l'utilisateur étant donné l'interaction de techniques différentes dans l'interface, du moins à l'heure actuelle. Par conséquent, l'interaction peut être source de frustrations pour l'utilisateur, particulièrement si le temps de réponse n'est pas uniforme pour des demandes similaires. L'utilisateur peut avoir une impression très nette de maîtrise dans un tel système. En effet, il existe bien des façons de diriger le système et de lui donner des instructions. Le système peut aider à guider l'interaction de l'utilisateur en surveillant la nature de cette interaction. Ainsi, il peut aider davantage l'utilisateur qui fait beaucoup d'erreurs ou encore

fournir moins de niveaux d'abstraction à l'utilisateur qui emprunte une voie qui lui est familière.

3.7 Jeux informatiques

Beaucoup de jeux informatiques ont été mis au point pour permettre l'accès à de l'information utile ou pour amuser les chercheurs. La plupart de ces jeux ne sont pas connus du grand public car ils utilisent des bases de données très spécialisées qui se trouvent dans des endroits très précis. Les laboratoires Bell, par exemple, ont mis au point un jeu que l'on nomme "EATS" et que l'on utilise pour trouver un restaurant d'un certain genre dans un endroit donné. Le principe de ce jeu consiste à chercher quelque chose en dialoguant avec le système. Pour chaque donnée que vous fournissez au système, celui-ci vous dit quelque chose sur votre situation actuelle. À partir de ce renseignement, vous pouvez dire si vous vous rapprochez ou non de votre but. Pour ces jeux, les techniques d'interaction reposent sur des questions posées par le système ou sur la présentation de choix d'actions par le système à l'intention de l'utilisateur.

Dans un tel contexte, les demandes sont spécifiées progressivement. Ainsi, dans un jeu du genre **question-réponse**, le système pose une question, l'utilisateur donne une réponse, puis le système renseigne l'utilisateur sur l'étape où en est rendu le jeu. Ce genre de formulation des demandes peut comporter l'utilisation de nombreuses frappes, mais les réponses sont habituellement courtes et peuvent n'exiger qu'un oui ou un non. Les commandes sont très simples et, une fois qu'elles sont données, le système guide l'utilisateur afin d'éviter des erreurs de syntaxe dans la demande. Cependant, on peut s'attirer des difficultés sur le plan de la sémantique, car les réponses aux questions sont données dans un contexte limité d'information sur les conditions d'utilisation du moment. Par conséquent, on peut facilement prendre une mauvaise direction. La sélectivité est bonne si l'on suit

une voie directe vers l'information désirée. Cependant, en cas de déviation, on peut obtenir des tas d'informations utiles. Malheureusement, la nature de la formulation de la demande dépend énormément de l'application. Par contre, les techniques utilisées peuvent être très insensibles au genre de base de données sous-jacente. L'interface peut facilement être personnalisée en fonction des applications particulières.

Les réponses sont présentées à l'utilisateur sous forme d'éléments. Une partie de la réponse peut lui être présentée ou il doit la découvrir à mesure que se fait la formulation de la demande. Par conséquent, il est normalement très facile de suivre la présentation puisque l'on ne reçoit que quelques faits à la fois. Les réponses sont normalement présentées sur un écran, mais il est possible d'utiliser des effets sonores pour indiquer que l'on s'approche du but ou que l'on est sur la bonne voie, comme c'est le cas pour les jeux vidéo. L'utilisateur peut régler le débit de sortie de la réponse en contrôlant la vitesse à laquelle il répond au système. En raison de la nature même de l'interaction, il peut être difficile de conserver et de réutiliser les réponses.

Pour ce qui est de la dynamique, l'interaction peut être très amusante pour l'utilisateur. L'intensité de l'entrée et celle de la sortie peuvent être assez grandes, surtout si les entrées sont réduites au minimum, dans le cas des oui et non, par exemple. Il est évident que l'interaction présente un haut niveau de ludicité. L'utilisateur participe à un jeu en vue de gagner quelque chose. Pour chaque jeu, il n'y a habituellement qu'un seul protocole d'interaction, mais ce protocole peut être différent pour chaque jeu. Étant donné le découpage court et progressif de l'interaction, les réponses, tant du système que de la part de l'utilisateur, peuvent être données très rapidement. Les réponses du système devraient arriver à un rythme uniforme puisque, à

Langages d'interrogation interactive pour bases de données externes

chaque étape, il ne peut y avoir qu'un nombre relativement faible de possibilités, particulièrement si les questions n'exigent qu'un oui ou un non pour réponse. L'utilisateur peut avoir ou non l'impression de maîtriser le système selon qu'il a l'impression de gagner ou de perdre. Si le jeu a été conçu de façon à gagner assez facilement, l'utilisateur aura alors l'impression d'exercer un contrôle sur le système.

THE UNIVERSITY OF CHICAGO
DIVISION OF THE PHYSICAL SCIENCES
DEPARTMENT OF CHEMISTRY
5708 SOUTH ELLIS AVENUE
CHICAGO, ILLINOIS 60637
TEL: (773) 835-3100
FAX: (773) 835-3101
WWW: WWW.CHEM.UCHICAGO.EDU

4 BESOINS EN MATIÈRE D'INTERROGATION

Dans une section précédente, nous avons divisé le processus d'interrogation en trois parties: la demande, la réponse et la dynamique de l'interaction. En ce qui concerne l'interaction utilisateur, c'est la demande et sa dynamique qui constituent la partie la plus difficile. En effet, lorsque l'utilisateur établit une demande, il essaie d'informer le système de ses besoins précis. Il doit fournir un grand nombre de renseignements et faire preuve d'une grande exactitude dans sa demande, s'il veut que le système l'interprète correctement. D'autre part, l'utilisateur a beaucoup plus de facilité à comprendre les réponses du système, même si la présentation de celles-ci laisse à désirer. Ceci s'explique du fait que les humains sont beaucoup plus intuitifs et qu'ils s'adaptent bien plus facilement que les systèmes informatiques. Cela ne signifie pas que les aspects de réponse à l'interrogation ne sont pas importants. Cependant, d'ici la fin du rapport, nous allons surtout nous pencher sur les aspects de demande du langage d'interrogation et de sa dynamique.

Tant pour les aspects de demande que pour les aspects de dynamique d'un langage d'interrogation, certaines caractéristiques ont été établies pour décrire les paramètres de l'interaction utilisateur. À partir de ces caractéristiques et de leur évaluation par rapport aux différents langages d'interrogation, nous pouvons fixer certaines caractéristiques dont devrait être doté, idéalement, un langage d'interrogation de base de données externes.

En général, la personne qui veut avoir accès à des bases de données externes ne sait pas dactylographier et ne veut pas être obligé de le faire pour être en mesure d'utiliser un langage d'interrogation. Donc, il faudrait qu'un nombre minimal de frappes permettent d'entrer en communication avec le système. Il faut absolument disposer d'un pointeur pour désigner les parties voulues de l'affichage. Ce pointeur

devrait être mécanique - un manche à balai ou une souris, par exemple - pour que l'utilisateur puisse se concentrer librement sur l'écran pendant qu'il s'en sert. Selon toute vraisemblance, il sera impossible d'éliminer le travail de dactylographie. Donc, il faut aussi que le système tolère et corrige les erreurs de dactylographie et que l'utilisateur puisse avoir recours à des pseudonymes pour faire des renvois à des objets dans des bases de données externes.

Toutes les commandes nécessaires pour indiquer au système qu'il doit intervenir doivent pouvoir être données au moyen de touches de fonction et de menus et être nommées d'après la fonction qu'elles permettent d'exécuter. En outre, il importe de restreindre à un minimum le nombre de ces commandes. Ainsi, l'utilisateur n'aura pas à se souvenir des commandes disponibles et pourra facilement se rappeler leurs fonctions d'après les noms.

Le système doit aider l'utilisateur à formuler sa demande. À cette fin, le système doit présenter les options offertes et, aussi, fournir des instructions en direct à tout moment au cours de l'interaction. La façon dont les demandes sont formulées doit faire appel à l'intuition de l'utilisateur. À cet égard, il devrait pouvoir disposer d'un paradigme facile à comprendre et qui l'aiderait à se souvenir des protocoles d'interaction. De plus, le système peut fournir des indices à l'utilisateur - au moyen de l'affichage, par exemple - qui l'aident à formuler correctement ses demandes.

En cherchant à aider l'utilisateur à formuler ses demandes, nous devons éviter de déterminer entièrement à l'avance les demandes qui peuvent être formulées, ce qui enlèverait toute liberté d'action à l'utilisateur. Le langage d'interrogation doit laisser une certaine latitude pour ce qui est de la sélectivité des demandes. À cette fin, le langage d'interrogation doit au moins permettre à l'utilisateur

de préciser le contenu de sa demande, si ce n'est sa forme exacte.

Dans la plupart des langages d'interrogation, l'utilisateur est censé connaître la structure ou le schéma de la base de données. Pour les bases de données externes, nous ne pouvons pas présumer que l'utilisateur connaîtra ou même comprendra la structure de la base de données ou la différence entre le schéma et la base de données. Il devrait plutôt y avoir une méthode permettant à l'utilisateur de passer facilement de la structure aux données. En outre, la façon dont l'utilisateur interroge la structure d'une base de données doit s'apparenter le plus possible à celle qu'il utilise pour interroger la base de données même. De cette façon, l'utilisateur n'a besoin d'apprendre qu'un seul jeu de protocoles d'interaction qui s'applique à toutes ses interactions avec le système.

Des études montrent que l'utilisateur du système informatique peut être fort désorienté s'il ne sait pas où il se trouve dans le système (Mantei, 1981). En outre, on ne peut s'attendre à ce qu'un utilisateur se souvienne de tous les éléments qui interviennent dans une interaction. Afin de donner à l'utilisateur des points de repère, nous devrions le plus possible conserver sur l'écran l'itinéraire suivi dans l'interrogation de la base de données (ou au moins en disposer pour en faire l'affichage au besoin).

Les interfaces d'utilisateur qui servent à consulter des données non mises en forme sont souvent fort différentes de celles qui servent à consulter des données mises en forme. Lorsque des bases de données externes sont interrogées, plusieurs types de données différents peuvent se présenter. La plupart des données se présenteront probablement sous une forme qui fera que l'utilisateur ne pourra pas distinguer les

données mises en forme (renseignements sur le stock, par exemple) des données non mises en forme (information sur les ventes par catalogue, par exemple). En outre, il n'est probablement pas souhaitable d'exiger un jeu distinct de protocole pour des types de données différents. Ainsi, l'interface de l'utilisateur doit être le plus uniforme possible pour l'interrogation des différents types de données que l'on trouve dans les bases de données externes.

Même si nous fournissons des protocoles d'interaction uniformes pour tous les types de demandes, nous pouvons admettre une interface d'utilisateur peu linéaire pour ce qui est de la difficulté éprouvée dans la formulation de la demande. Autrement dit, il est très facile de formuler des demandes simples, tandis qu'il peut être extrêmement difficile de formuler des demandes plus compliquées. En fait, on peut supposer que des sommes logiques et autres demandes complexes ne se présentent pas fréquemment. Quoi qu'il en soit, c'est la performance qui laissera à désirer lorsque nous exécuterons ces opérations. Pour que l'utilisateur puisse faire ces interrogations, il lui faudra une interface évoluée et spécialisée qui donne accès aux mêmes données que l'interface pour non-spécialistes.

Finalement, l'interface d'utilisateur devrait tirer profit de l'aptitude des gens à se souvenir d'abstractions. En outre, on devrait avoir recours à la technologie disponible (le graphisme et la couleur) pour faire la distinction entre les différents types d'objets visualisés à l'écran. Par exemple, nous devrions faire la distinction entre les données que l'utilisateur recherche, les données que l'on peut choisir et celles qui sont fournies par le système. Nous n'insisterons pas davantage sur cet aspect, mais il peut revêtir beaucoup d'importance puisqu'il guide l'utilisateur dans ses interactions avec le système. Il faut donc lui accorder une attention spéciale au moment de décider de la disposition de l'affichage qui sera présentée à l'utilisateur.

Langages d'interrogation interactive pour bases de données externes

À partir de ces considérations comme principe directeur de la formulation des demandes et de l'interaction utilisateur, nous allons esquisser un langage d'interrogation des bases de données externes dans les pages qui suivent. Nous présentons d'abord, pour le langage d'interrogation, un paradigme qui véhicule par un exemple facile à comprendre l'essence de l'interaction de l'utilisateur, ensuite une vue d'ensemble de la conception d'un langage d'interrogation et finalement, une analyse des facteurs de conception de ce langage.

THE UNIVERSITY OF CHICAGO
DIVISION OF THE PHYSICAL SCIENCES
DEPARTMENT OF CHEMISTRY
5708 SOUTH ELLIS AVENUE
CHICAGO, ILLINOIS 60637
TEL: 773-936-3700
WWW.CHEM.UCHICAGO.EDU

5 PARADIGME

Pour avoir accès à des informations au moyen d'un langage d'interrogation interactive de bases de données externes, l'utilisateur doit disposer d'un paradigme qu'il peut suivre et comprendre facilement. Ce paradigme doit aider l'utilisateur à se souvenir des protocoles d'interrogation (et à les connaître). Le paradigme que nous avons choisi est celui de la navigation d'un vaisseau spatial personnel dans l'espace.

Notre vaisseau spatial est très puissant et sa conception est des plus récentes. Il nous permet de voyager avec une facilité relative tant sur de longues distances - entre les galaxies - que sur de courtes distances - entre les planètes voisines. Le but de notre voyage dans l'espace est de parvenir à une planète et de l'explorer, quoique nous puissions aussi nous borner à une exploration de l'espace en soi. Une fois parvenus à une planète, nous pouvons l'explorer en la survolant ou en nous y posant et en examinant plus en détail une région donnée.

L'espace est très vaste et complexe et ne nous est généralement pas familier. Notre vaisseau est équipé d'un écran qui nous permet de voir l'espace dans toutes les directions, mais seulement une partie à la fois. Il est très difficile de naviguer sans aides, c'est-à-dire sans instruments, quoiqu'il puisse être préférable de procéder ainsi lorsqu'on explore une partie de la planète. Pour faciliter nos déplacements, notre vaisseau est équipé d'instruments qui peuvent dessiner des cartes dans le détail. Ces cartes peuvent prendre différents niveaux d'abstraction: structure générale d'une galaxie ou structure générale d'une planète, avec niveaux intermédiaires d'abstraction. On peut les projeter sur l'écran de visualisation du vaisseau.

Nous pouvons naviguer de plusieurs façons. Si nous avons une destination en vue, nous pouvons la préciser et notre vaisseau nous y amènera automatiquement. D'autre part, si nous ne faisons qu'explorer l'espace, il se peut que nous ne sachions pas exactement quelle sera notre prochaine destination. Nous pouvons nous en faire une idée ou accorder de l'intérêt à toute exploration interplanétaire. Si nous voulons savoir quelles destinations sont à notre portée, nous n'avons qu'à lire les cartes présentées sur l'écran de visualisation. Nous pouvons demander l'affichage de certaines cartes présentant une ou quelques propriétés particulières, ou toutes les cartes. En général, au fur et à mesure que nous envisageons les différentes destinations possibles, nous consulterons plusieurs cartes, des plus générales aux plus précises (en consultant, dans cet ordre, des cartes de la galaxie, des étoiles, puis des planètes). Ensuite, nous choisirons une destination et nous programmerons les instruments pour que notre vaisseau nous y conduise.

De la même façon nous pouvons imaginer qu'il existe des sources d'information auxquelles nous désirons avoir accès et qui se trouvent dans un **espace informationnel**. La composition de cet espace informationnel est modelée sur celle de l'espace même. Cet espace se compose d'objets abstraits comme des galaxies et des systèmes stellaires et d'objets concrets comme les planètes. Les objets abstraits de l'espace informationnel correspondent à la **structure** (schéma) des données. La structure nous renseigne d'une certaine façon sur la nature des données auxquelles nous pouvons avoir accès. La partie concrète correspond aux **données** (bases de données) de l'espace informationnel. La partie "données" correspond au niveau le plus bas des détails que nous pouvons explorer dans l'espace informationnel.

Grâce à notre vaisseau (système informatique), nous pouvons explorer l'espace informationnel. Le poste de commandement de notre vaisseau est constitué par le langage d'interrogation et notre écran de

visualisation par l'écran cathodique (de télévision) qui nous permet d'observer l'espace informationnel. Tout comme nous ne pouvons voir qu'une partie de l'espace en un temps donné sur l'écran de visualisation de notre vaisseau, notre écran cathodique ne nous présente qu'une partie de l'espace informationnel à un moment précis. L'écran cathodique définit les limites de l'espace informationnel qui sont observables comme l'écran de visualisation de notre vaisseau définit la partie de l'espace visible. Nous dirons de l'écran cathodique que c'est notre **portique** sur l'espace informationnel puisqu'il sert de fenêtre d'observation et nous permet de voir uniquement les informations auxquelles la fenêtre donne accès.

Les manoeuvres dans l'espace informationnel reposent sur trois actions essentielles, comme pour les manoeuvres faites pour se déplacer dans l'espace extra-atmosphérique. Tout d'abord, nous devons pouvoir afficher dans notre portique les structures (cartes) de l'espace informationnel qui nous intéressent, afin de choisir notre destination. Deuxièmement, nous devons pouvoir nous rendre à destination, c'est-à-dire avoir accès à la source d'information qui présente un intérêt. Finalement, nous devons être en mesure d'explorer les données qui se trouvent dans la source d'information choisie.

Deux remarques s'imposent au sujet de ce paradigme. D'une part, nous séparons distinctement la structure des données sur le plan conceptuel mais, sur le plan opérationnel, (comme nous allons le voir), structure et données sont traitées de la même façon (ou presque). Cette séparation pourrait se révéler importante à l'avenir puisque la structure pourrait résider dans le terminal de l'utilisateur tout comme les cartes précédentes sont gardées dans le vaisseau spatial. Les données se retrouvent dans une source d'information et, pour y avoir accès et les explorer, l'utilisateur doit s'y rendre tout comme il doit se rendre sur une planète pour l'explorer en détail. D'autre part, par la façon dont nous avons décrit l'interaction de l'utilisateur avec le

système, la plupart des modalités d'accès à l'information se règlent automatiquement. L'utilisateur fournit certains points de repère, mais c'est le système qui décide de la façon dont les choses se déroulent. L'accès automatique est préférable pour le profane, mais le spécialiste pourrait le trouver ennuyeux. L'utilisateur pourrait prendre en main les commandes et diriger le système plus directement, tout comme nous pourrions diriger notre vaisseau manuellement. L'utilisateur expérimenté disposerait ainsi d'un langage d'interrogation plus puissant et plus complexe. Dans ce rapport, nous examinons uniquement le langage d'interrogation dit automatique, mais il ne faut pas qualifier d'impossible l'élaboration d'un langage d'interrogation "manuel" qui compléterait le précédent.

6 VUE D'ENSEMBLE DE LA CONCEPTION

La présente section, décrit la conception globale d'un langage d'interrogation interactif servant à donner accès à des bases de données externes. Nous allons voir **ce qui** peut être fait et non **la façon** de le faire. Ainsi, nous allons décrire les étapes de façon générique et utiliser des terme génériques comme "pointer", "remplir" et "choisir" sans tenir compte de la façon dont ces actions se font. Nous aurons recours à l'analogie établie dans la section précédente pour expliquer les opérations en langage d'interrogation. Nous allons examiner ensuite quelques-uns des moyens dont dispose l'utilisateur pour exécuter ces opérations.

6.1 Choix d'une base de données externe

Supposons que nous sommes dans notre vaisseau spatial quelque part dans l'espace et que nous voulons choisir une destination. À l'aide de notre écran de visualisation, nous pouvons faire un balayage de l'espace et observer les étoiles. Cependant, tout ce que cela nous permet de voir, c'est une représentation abstraite. Le pilote chevronné peut choisir une destination directement à partir de la position des étoiles et mettre le cap sur cette destination. Cependant, en général, nous ne serons probablement pas familiarisés avec l'espace. Pour choisir une destination, nous pouvons faire le balayage de différentes parties de l'espace et consulter des cartes représentant chacune de ces parties. Nous pouvons demander à voir certaines parties de l'espace qui présentent des propriétés particulières ou nous pouvons observer toutes les parties de l'espace. Pendant la consultation des cartes, nous pouvons demander une autre carte plus détaillée d'une région donnée. Nous pouvons aussi faire des annotations dans le cas de certaines régions qui présentent un intérêt particulier et sur lesquelles on voudrait revenir. Plus tard, nous déciderons de la destination avant de fixer le cap.

Imaginons maintenant que nous essayons de consulter une base de données externe. La première fois que nous entrons en communication avec le système, nous ressemblons au passager du vaisseau spatial. Nous sommes dans l'espace extra-atmosphérique et, du portique, nous pouvons voir une représentation abstraite (modèle) des bases de données qui nous sont offertes. L'image est comme le balayage de l'espace qui nous montre simplement les configurations stellaires sans autre information sur ce qu'elles représentent. Par exemple, lorsque nous sommes dans le portique, nous pouvons observer un modèle semblable à celui de la figure 1.

Le format exact du modèle dans le portique ne revêt pas d'importance ici (une étude permettrait de déterminer la meilleure présentation). Il importe de retenir que le modèle comprend trois parties essentielles. Une de ces parties précise le **trajet** suivi pour arriver à ce modèle dans le portique. Dans notre exemple, ce renseignement se retrouve dans la zone ITINÉRAIRE. Une deuxième partie sert à la **spécification** et à **l'affichage des données**. Dans notre exemple, ces renseignements correspondent aux zones CATÉGORIE et CONTENU. Finalement, il y a une partie réservée à la **rétroaction du système** que nous avons nommée MESSAGE.

SERVICE D'INFORMATION VIDÉOTEX

ITINÉRAIRE: _____

CATÉGORIE	CONTENU
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

MESSAGE: _____

Figure 1: Exemple de modèle

Si nous ignorons à quelle base de données externe nous voulons avoir accès, nous pouvons demander au système d'afficher tous les renseignements dont il dispose sur les bases de données à ce niveau. Nous obtenons alors l'affichage de toutes les entrées qui correspondent au modèle. Par exemple, nous pourrions obtenir un affichage comme celui illustré à la figure 2 si nous avons demandé l'affichage de toutes les entrées.

SERVICE D'INFORMATION VIDÉOTEX

ITINÉRAIRE: _____

CATÉGORIE	CONTENU
Divertissements	Films Spectacles Expositions Sports Télévision
Nouvelles	Nouvelles - internationales - nationales - régionales Nouvelles du sport Divertissements
Restaurants	Météo Chinois

MESSAGE: Première de dix images

Figure 2: Exemple d'affichage

Nous pouvons maintenant examiner la carte d'information, et marquer certaines entrées pour référence ultérieure en les pointant ou en choisissant une par pointage. Le choix d'une entrée nous permet d'avoir accès à des renseignements plus précis. Par exemple, si l'on choisit la catégorie "divertissements" et si l'on demande l'affichage de toutes les entrées comprises dans cette catégorie, nous obtiendrons un affichage comme celui de la figure 3.

Langages d'interrogation interactive pour bases de données externes

Il se peut aussi que nous soyons intéressés par certaines bases de données, mais que l'on en ignore l'emplacement. Dans ce cas, nous pouvons demander au système de faire apparaître plus d'informations sur ces destinations en pointant et en remplissant la zone CATÉGORIE. Par exemple, nous pouvons introduire l'élément "divertissements" dans la zone CATÉGORIE et en demander l'affichage du contenu comme à la figure 4.

SERVICE D'INFORMATION VIDÉOTEX

INTINÉRAIRE: Divertissements

CATÉGORIE	CONTENU
Films	Premières Revue Festivals
Spectacles	Théâtre Ballet Opéra Récitals
Expositions	Art Livres Artisanat
Sports	Hockey Soccer

MESSAGE: Première de deux images

Figure 3: Exemple d'affichage

Par la suite, nous pouvons examiner la carte d'information, marquer certaines entrées pour référence ultérieure et(ou) en choisir une en la

pointant. Par exemple, si nous choisissons l'entrée "divertissements" et demandons par la suite l'affichage des entrées, nous obtenons l'affichage illustré à la figure 3.

Plutôt que d'entrer de façon explicite une valeur de zone, nous pouvons indiquer une valeur en pointant la zone et en demandant l'affichage de toutes les valeurs. Nous pouvons alors considérer ces entrées comme éléments de remplissage de cette zone. Dans ce cas, plutôt que d'inscrire une valeur de façon explicite, nous le faisons de façon implicite en identifiant certaines valeurs parmi celles qui sont affichées. Ainsi, par exemple, nous pourrions préciser l'affichage illustré à la figure 4 en pointant et en demandant l'affichage des valeurs de la zone CATÉGORIE, en marquant l'entrée "divertissements" et en demandant un nouvel affichage.

SERVICE D'INFORMATION VIDÉOTEX

ITINÉRAIRE: _____

CATÉGORIE	CONTENU
Divertissements	Films
	Spectacles
	Expositions
	Sports
	Télévision
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

MESSAGE: Première et seule image

Figure 4: Exemple d'affichage

Finalement, si nous savons où nous voulons nous rendre, nous pouvons l'indiquer directement et le système se chargera de nous y amener. Pour ce faire, nous pointons une valeur pour la zone ITINÉRAIRE et nous l'y faisons entrer, ce qui indique au système qu'il doit mettre le cap sur la destination choisie et nous présenter un modèle de celle-ci.

Par des applications répétées de cette technique, nous pouvons naviguer dans la structure d'une base de données et arriver à l'information recherchée. Chaque fois que nous choisissons un élément

dans un modèle, nous obtenons des renseignements plus détaillés sur l'itinéraire conduisant à une base de données. Nous pouvons également conserver de l'information sur le trajet choisi pour en arriver au modèle en cours. Ceci revient à essayer de déterminer une destination pour notre vaisseau spatial. On peut avoir recours à cette technique à plusieurs niveaux, mais il est souhaitable que le nombre des niveaux soit restreint pour réduire la quantité d'interactions requises pour en arriver aux données. En général, le nombre de niveaux requis est déterminé par la complexité de la base de données à laquelle nous voulons avoir accès. Cet aspect correspond au but que nous nous sommes fixé d'assortir la complexité de l'accès à la complexité de l'information ou du type de demande formulée.

Au moyen des zones CATÉGORIE et CONTENU, nous nous ménageons en fait la possibilité d'utiliser des mots clés et des renvois. Autrement dit, toutes les bases de données ayant trait à une ou à des catégories déterminées seront affichées. De même, certaines données peuvent se retrouver dans plusieurs catégories. On note que les données affichées peuvent se retrouver à différents niveaux dans la structure. Autrement dit, certains peuvent renvoyer à des bases de données proprement dites, tandis que d'autres peuvent se rapporter à des parties d'une base de données. Cependant, l'utilisateur n'a pas besoin de connaître cette distinction. Ce qui apparaît dans le portique est fonction de la façon dont les renvois aux données ont été conçus dans le système.

Chaque mot-clé affiché dans le portique est accompagné d'une description plus détaillée de l'information représentée par ce mot. À tout moment, nous pouvons pointer un mot-clé et en demander la description, tout comme on peut consulter le guide qui accompagne une carte. Nous procédons ainsi uniquement lorsque nous voulons en connaître davantage sur un point donné.

La zone ITINÉRAIRE nous permet de voir le trajet suivi pour en arriver à l'emplacement actuel. On peut revenir sur ses pas en tout temps en pointant le niveau (trajet) où nous voulons nous retrouver.

6.2 Exploration des données

Supposons que notre vaisseau se dirige vers une planète. Une fois arrivés, nous pouvons survoler la planète et en faire l'observation ou nous y poser et en explorer une partie. Pour nous aider à naviguer, nous avons besoin de cartes. Ces cartes montrent la structure abstraite de la planète, sans entrer dans les détails. Nous pouvons explorer la planète en détail en la survolant ou en nous y posant.

De la même façon nous en venons à atteindre la partie "données" d'une base de données. Tout comme nous explorons et nous survolons la structure, nous pouvons maintenant explorer et survoler la partie "données". Comme dans le cas de la structure, une représentation abstraite (modèle) des données du portique nous est présentée ici. Un exemple de ce modèle apparaît à la figure 5.

SERVICE D'INFORMATION VIDÉOTEX

ITINÉRAIRE: Divertissements. Films. Premières

FILMS	VEDETTE	CINÉMA	HEURES	TÉLÉPHONE
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____

MESSAGE: _____

Figure 5: Exemple de modèle

Nous choisissons des entrées de modèle particulières en pointant et en remplissant une zone. Par exemple, pour obtenir l'affichage de tous les films qui mettent en vedette Donald Sutherland, il suffirait de pointer la zone VEDETTE et d'y inscrire (la valeur) "Donald Sutherland". Le système nous indiquera alors tous les films qui mettent en vedette Donald Sutherland et qui sont à ce moment-là à l'affiche dans les cinémas. On peut également remplir une zone en la pointant et en demandant au système d'afficher toutes les valeurs qu'il possède pour

cette zone. Nous pouvons alors marquer certaines valeurs entrant dans cette zone. Le système nous donnera la liste complète de toutes les entrées où apparaît cette valeur. Par exemple, nous pourrions demander à voir toutes les valeurs correspondant à la zone VEDETTE. Nous n'aurions ensuite qu'à retenir pour examen ultérieur les valeurs qui nous intéressent.

Si nous ne voulons que parcourir un document, nous pouvons alors demander l'affichage de toutes les entrées du modèle. Par la suite, nous pouvons marquer certaines entrées particulières en les pointant, pour y revenir et en faire l'examen. Ceci permet d'éliminer les entrées qui ne nous intéressent pas et d'économiser l'espace d'affichage.

6.3 Itinéraires préétablis

Au cours de notre voyage en vaisseau spatial, nous découvrons sans tarder qu'il y a certaines destinations très fréquemment visitées. Nous en venons à connaître les itinéraires et nous pouvons rapidement nous passer de cartes. Une navigation à vue est possible.

De la même façon, lorsque nous avons accès à des bases de données externes, il y a certaines données que nous consultons très fréquemment. Pour accélérer les choses, nous pouvons établir à l'avance l'itinéraire que le système devra suivre. Nous pouvons nommer cet itinéraire et le préciser de façon explicite dans la zone ITINÉRAIRE, tout comme nous pointons directement notre destination sur les configurations stellaires visuelles que nous apercevons de notre vaisseau.

Ces itinéraires préétablis sont annexés à la structure du service d'information vidéotex, mais seul l'utilisateur qui les a définis y a accès. La destination peut être un point de la structure, les données ou une portion des données, c'est-à-dire un sous-ensemble de données.

Langages d'interrogation interactive pour bases de données externes

De cette façon, on peut, dans une certaine mesure, personnaliser le résultat au profit de l'utilisateur en éliminant certains renseignements qui se trouvent dans la partie donnée, par exemple. On peut distinguer les noms des itinéraires préétablis du nom des bases de données en utilisant des moyens appropriés pour les mettre en évidence.

7 CONSIDÉRATIONS EN MATIÈRE DE CONCEPTION

Dans la section précédente, nous avons vu ce que le langage d'interrogation peut faire, mais non pas de quelle façon il peut le faire. Nous avons eu recours à des termes génériques comme "pointer", "remplir" et "choisir" sans préciser de quelle façon ces actions doivent être faites. Dans la présente section, nous verrons plus en détail comment un utilisateur du service vidéotex peut réaliser ces actions.

7.1 Pointage

Une des actions les plus fondamentales dans les protocoles d'interaction que nous avons vus, c'est la possibilité de pointer un élément dans un portique en vue de le marquer ou de le choisir pour des actions ultérieures. Idéalement, ce pointage ne doit pas détourner l'attention de l'utilisateur de l'affichage dans le portique. Le pointage sonore serait de loin le plus souhaitable, mais il n'est peut-être pas encore réalisable sur les plans technique ou économique. Sur le plan qualitatif, le deuxième pointeur serait un pointeur mécanique comme une manche à balai, une rondelle ou une souris, l'utilisateur pouvant alors positionner un curseur à l'endroit voulu dans le portique et indiquer, d'une certaine façon, qu'il choisit cette position. (À noter qu'il est nécessaire de positionner les curseurs, mais aussi d'indiquer que c'est la position voulue.) Cette méthode de pointage est réalisable sur le plan technique et pourrait être rentable.

Faute d'avoir un pointeur pour préciser un emplacement arbitraire dans le portique, on peut toujours préciser des emplacements donnés, c'est-à-dire des zones particulières dans le portique. Cette précision devrait se faire aussi facilement que possible et tenir compte du fait que le positionnement est fonction de la structure du modèle affiché

dans le portique. Nous avons généralement besoin de pouvoir préciser une zone dans un modèle et peut-être une entrée dans une zone. Pour des raisons de généralité, ces deux précisions doivent être indépendantes l'une de l'autre. La sémantique exacte de la précision dépendra des caractéristiques du modèle affiché dans le portique.

La possibilité de choisir une zone et une entrée particulières conduit à la notion de zone et d'entrée "courantes". De façon visuelle, cette situation peut être représentée en esquissant à grands traits l'entrée et la zone courantes d'une certaine façon, c'est-à-dire par le clignotement, l'inversion de l'image ou l'utilisation d'une couleur donnée, par exemple. On peut préciser des opérations qui s'appliquent ensuite à l'entrée ou à la zone courantes comme nous verrons bientôt.

Pour la spécification des zones, il existe une opération de **zone** qui présente les options et la sémantique suivantes: **zone suivante** - permet de choisir la zone suivante; **zone précédente** - permet de choisir la zone précédente; **nom** de la **zone** - permet de choisir la zone dénommée.

Les zones d'un modèle comportent un ordre préétabli qui est utilisé par les options **zone suivante** et **zone précédente**. En outre, les zones se bouclent. La zone initiale est constituée par l'ensemble du modèle comme la zone qui vient après la dernière zone et avant la première. L'option **nom** fait appel à un nom symbolique pour faire directement référence à une zone. Il faut établir des conventions pour nommer les zones de façon symbolique. Certaines fonctions (définies ci-dessous) ont besoin d'une zone comme paramètre. Si aucune zone n'est précisée de façon explicite, c'est l'ensemble du modèle qui joue ce rôle. Ainsi, lorsqu'une zone n'est pas expressément définie, le mode **affichage** renvoie à l'affichage de toutes les valeurs d'un modèle,

tandis que le mode **affichage** qui suit le choix d'une zone renvoie au seul affichage des valeurs correspondant à la zone choisie.

Pour la spécification des entrées, il existe une opération **entrée** qui permet de choisir une entrée d'une zone à l'aide des options et de la sémantique suivantes: **entrée suivante** - permet de choisir l'entrée suivante; **entrée précédente** - permet de choisir l'entrée précédente: **nom d'entrée** - permet de choisir l'entrée appelée.

Les entrées présentent un ordre préétabli qu'utilisent les options **entrée suivante** et **entrée précédente**. On peut aussi faire référence à des entrées au moyen de noms symboliques qui permettent de choisir directement une entrée. Par exemple, pour une liste des valeurs d'une zone, on peut désigner symboliquement les entrées par un, deux, etc. Si la zone courante est constituée par le modèle, une entrée renvoie alors à toutes les valeurs qui définissent un des cas du modèle.

7.2 Positionnement dans le portique

Jusqu'à maintenant, nous n'avons tenu compte d'aucune restriction quant aux dimensions du modèle qui pourrait être affiché dans le portique. Nous avons supposé que tout élément pouvait être affiché immédiatement. Cependant, il se peut que les modèles soient trop encombrants pour entrer dans le portique. Ainsi, les dimensions de notre portique restreignent le nombre de données que nous pouvons afficher en un temps donné. Idéalement, nous aimerions pouvoir déplacer le portique continuellement pour suivre le modèle comme s'il s'agissait d'une caméra de télévision.

Dans les systèmes vidéotex, l'espace informationnel est habituellement divisé en unités de dimensions fixes (pages) correspondant à celles du portique (écran de télévision) et un mouvement

continu dans toutes les directions est impossible. Par conséquent, nous pouvons supposer que les modèles sont plutôt divisés en **images de dimensions** fixes. Une image est une partie de l'espace informationnel que nous pouvons observer dans le portique à un moment donné (tout comme on peut regarder une diapositive à l'aide d'un projecteur). En général, on peut, à partir d'une image, passer à n'importe quelle autre image (contrairement à ce qui se produit dans le cas d'une projection de diapositives où seules la diapositive précédente ou la diapositive suivante peuvent être vues).

Une image renferme uniquement le nombre de données nécessaires pour remplir le portique. Les images qui se trouvent au-dessus et au-dessous de l'image observée contiennent, respectivement, les entrées de modèle précédentes et suivantes. Celles de droite et celle de gauche contiennent des entrées supplémentaires qui ont trait à l'image observée, mais qui ne peuvent être affichées immédiatement dans le portique. Ainsi, par exemple, des tables de données trop larges pour entrer dans le portique peuvent être réparties en plusieurs images adjacentes verticales (à la droite et(ou) à la gauche l'une de l'autre). Des tables de données trop longues pour entrer dans le portique peuvent être divisées en plusieurs images adjacentes horizontales (l'une au-dessus (ou) au-dessous de l'autre).

Pour amener le portique en position avec l'image voulue, il existe une opération **image** avec les options et la sémantique suivantes: **image supérieure** - permet de déplacer le portique jusqu'à l'image qui se trouve au-dessus de l'image observée; **image inférieure** - permet de déplacer le portique jusqu'à l'image qui se trouve au-dessous de l'image observée; **image de gauche** - permet de déplacer le portique jusqu'à l'image qui se trouve à la gauche de l'image observée; **image de droite** - permet de déplacer le portique jusqu'à l'image qui se trouve à la droite de l'image observée; **nom de l'image** - permet de déplacer le portique jusqu'à l'image appelée.

On peut, symboliquement, donner un nom aux images pour qu'il soit possible d'amener directement le portique sur ces images. Dans la plupart des cas, et probablement au début, il est souhaitable de ne pas toucher aux images de droite et de gauche puisque ceci aurait pour effet de fragmenter considérablement les données. Dans ce cas, la fonction **image** se réduit aux fonctions **image supérieure**, **image inférieure** et **nom de l'image**.

S'il est nécessaire de faire intervenir les images de gauche et de droite, il est alors souhaitable d'avoir recours à une certaine méthode pour faire connaître à l'utilisateur sa position (l'orienter). Cette orientation comporte deux aspects. Tout d'abord nous cherchons à informer l'utilisateur sur sa position par rapport à l'ensemble de l'espace informationnel visible à ce niveau. Une technique utilisée à cette fin consiste à illustrer graphiquement la position du portique par rapport à l'ensemble de l'espace informationnel. Par exemple, nous pourrions montrer l'espace informationnel dans son ensemble sous la forme d'un polygone et la position du portique sous la forme d'un polygone superposé au polygone plus grand qui représente l'espace informationnel. Comme autre solution, nous pourrions utiliser la zone du message pour indiquer le déplacement de l'utilisateur vers la gauche ou vers la droite par rapport à un point de référence donné, comme nous avons illustré le déplacement à la verticale dans les exemples précédents.

Le deuxième aspect qui doit nous préoccuper, c'est le maintien d'un lien sémantique entre les différentes images qui se trouvent dans le portique. Dans le plan vertical, la question ne se pose pas vraiment, étant donné qu'en faisant un balayage ascendant et descendant, la même partie du modèle est affichée dans le portique. Cependant, dans le plan horizontal, la partie du modèle visible dans le portique change au fur et à mesure de nos déplacements vers la gauche et vers la droite. Il est donc souhaitable de conserver un certain point commun entre les

images adjacentes horizontales. Pour ce faire, on peut choisir une des zones de modèle comme **zone clé** et la garder dans le portique pendant toute la durée de notre déplacement horizontal.

Par exemple, à la figure 5, une des zones du modèle, disons la zone FILMS, pourrait demeurer dans le portique pendant que nous déplaçons la position du portique dans le plan horizontal.

7.3 Fonctions de commande

Les fonctions de commande servent à signaler au système qu'il doit fournir des renseignements sur son état et prendre des mesures appropriées. Le recours à une fonction de commande peut être vu comme le recours à une procédure qui prend comme entrée l'état ponctuel du système et(ou) la valeur de la zone. La valeur des paramètres d'entrée donne lieu à certaines actions. Les fonctions de commande peuvent prendre différentes formes, par exemple, boutons de commande ou commandes de menu.

CHOIX

La fonction **choix** prend comme paramètre d'entrée la valeur de la zone courante. Le résultat de la fonction dépend des caractéristiques de la valeur de la zone. Si nous choisissons une valeur de la zone "catégorie", celle-ci devient partie intégrante de l'itinéraire que nous avons emprunté. Si nous choisissons un nom dans la zone ITINÉRAIRE, le système nous ramène à ce point. Comme nous allons le voir, si nous choisissons une valeur de données qui est du type **image** ou **son**, nous obtenons l'affichage d'une image, la présentation d'un film ou d'une bande magnétoscopique ou la lecture de messages sonores enregistrés. S'il n'existe pas de procédure explicite définie pour une valeur de zone, le fait de la choisir ne produit aucun effet et il en résulte donc une action nulle.

REPÉRAGE

La fonction **repérage** prend comme paramètre d'entrée la valeur de la zone courante. Cette fonction donne au système l'ordre de prendre note de la valeur repérée pour référence ultérieure. Par exemple, lorsque nous examinons des valeurs de zone, nous pouvons repérer certaines zones pour en faire l'examen ultérieur. Le repérage ne peut se faire qu'en mode de spécification **d'affichage**.

DESCRIPTION

La fonction **description** prend comme paramètre d'entrée la valeur de la zone courante. Le système répond à cette fonction en fournissant une description détaillée de la valeur de la zone courante. Cette explication peut être visuelle, verbale ou prendre ces deux formes en même temps. Par exemple, pour une valeur de zone **catégorie**, la fonction **description** peut servir à obtenir une explication des renseignements qui se trouvent dans la catégorie. Les valeurs de zone n'exigent pas toutes une description. Donc, il n'y aura pas nécessairement de description lorsqu'on fait appel à cette fonction, c'est-à-dire qu'il y aura une action nulle.

AIDE

La fonction **aide** prend comme paramètre d'entrée l'état ponctuel du système. S'il fait appel à cette fonction, l'utilisateur reçoit des instructions sur les mesures qu'il peut prendre et sur la façon dont il peut les prendre compte tenu de l'état ponctuel du système.

ANNULATION

La fonction **annulation** prend comme paramètre d'entrée l'état ponctuel du système. Cette fonction donne au système l'ordre d'annuler la dernière commande ou le mode de spécification en cours.

INTERRUPTION

La fonction **interruption** prend comme paramètre d'entrée l'état ponctuel du système. Cette fonction conserve l'état ponctuel de l'interaction utilisateur. On peut relancer l'interaction utilisateur n'importe quand. L'interaction reprend alors à partir du point d'interruption, comme si elle n'avait jamais été interrompue. Les interactions interrompues sont conservées dans la base "interruption". On peut mettre fin à une interruption en choisissant ces interactions.

SORTIE

La fonction **sortie** prend comme paramètre d'entrée l'état ponctuel du système. Cette fonction met fin à l'interaction utilisateur en cours et aucune information sur cette interaction n'est conservée.

CONSERVATION

La fonction **conservation** prend comme paramètres d'entrée l'état ponctuel du système plus un nom fourni par l'utilisateur. Cette fonction conserve la spécification de l'interaction utilisateur et la dénomme de façon explicite. On peut ensuite copier l'interaction en choisissant le nom défini par l'utilisateur. Ce nom devient partie intégrante de l'information de la structure pour cet utilisateur. La fonction **conservation** permet de définir ce que l'on voit. Elle diffère de la fonction **interruption** - puisqu'une interaction interrompue n'est conservée que jusqu'à ce qu'elle soit achevée.

D'autre part, une interaction conservée est gardée en permanence par le système. On peut laisser tomber une interaction conservée en gardant un état de système nul (initial) avec le même nom que l'état courant de système conservé.

7.4 Modes de spécification

Les modes de spécification ont trait à la façon dont un utilisateur spécifie des demandes de renseignements adressées au système. Lorsqu'on fait appel à un mode de spécification, une suite d'actions va suivre pour constituer la spécification. On peut mettre fin implicitement à un mode de spécification au moyen de la fonction **choix**, de la fonction **conservation** ou d'un autre mode de spécification, ou explicitement par la fonction **annulation** ou la fonction **sortie**. Pendant l'interaction avec des bases de données externes, l'utilisateur peut spécifier sa demande de deux façons. L'une de ces méthodes est le mode par défaut; c'est l'utilisateur qui précisera lequel. Le mode par défaut est introduit initialement lorsque l'utilisateur entre en communication avec le système et après chaque fonction **choix** ou **conservation**.

REPLISSAGE

Le mode **remplissage** permet à l'utilisateur de choisir diverses zones et de préciser la ou les valeurs pour chacune. Il choisit les zones en les pointant. Ces dernières sont remplies explicitement au moyen d'une valeur qu'on y inscrit à partir d'un clavier ou, implicitement, en marquant des valeurs affichées comme on l'explique au mode **affichage**.

Si plusieurs valeurs sont précisées pour une zone donnée, on suppose alors que l'opération booléenne **ou** fait partie de ces valeurs (correspondance pour valeur 1 ou valeur 2, entre autres). Il est à noter qu'une opération booléenne **et** parmi ces valeurs est

impossible sauf si une entrée d'une zone peut effectivement prendre plusieurs valeurs. Si plus d'une zone est remplie, on suppose que l'opération booléenne et fait partie des zones choisies (correspondance pour zone 1 et zone 2, entre autres). On peut exécuter une opération booléenne ou comprise dans les zones si l'on remplit plusieurs modèles.

On indique la fin du remplissage d'un modèle en précisant le mode **affichage**, le modèle étant constitué par la zone en cours. L'utilisateur perçoit alors toutes les entrées du modèle qui correspondent à la spécification. Si aucune zone n'a été remplie, toutes les entrées correspondant au modèle sont affichées.

AFFICHAGE

Le mode **affichage** permet à l'utilisateur de voir toutes les entrées des zones ou des modèles dans le portique. Si le modèle est vide au moment où l'on fait intervenir le mode **affichage**, toutes les entrées sont affichées. Si le modèle est rempli, seules les entrées qui correspondent au modèle rempli sont affichées comme on l'indique au paragraphe où l'on décrit le mode de spécification **remplissage**.

En mode **affichage**, les entrées de modèle peuvent être explorées au moyen des opérations de pointage et de positionnement dans le portique. On peut choisir la zone et l'entrée courantes au moyen de la fonction **choix** qui fait appel à la procédure correspondant à la valeur de la zone en cours. On peut aussi marquer d'un repère les valeurs des zones en mode **affichage**. Consécutivement, si l'on passe de nouveau en mode **affichage**, seules les valeurs des zones marquées d'un repère seront choisies et affichées. Si la zone en cours est le modèle, seules les entrées repérées seront choisies en vue d'être affichées. Si la zone courante est une zone qui se trouve dans le modèle, seules les entrées qui correspondent aux valeurs des zones

repérées seront choisies. Les conventions du mode **remplissage** s'appliquent ici.

7.5 Types de données

La plupart des langages d'interrogation traditionnels ne traitent raisonnablement que des données mises en forme. Cependant, il existe d'autres types de données que l'on peut avoir à traiter lorsqu'on interroge des bases de données externes. En voici quelques-uns: données de textes mises en forme, données de textes non mises en forme, données d'images et données vocales.

En raison de la nature des différents types de données, le mode d'interrogation est propre à chaque type de données. Cependant, on peut utiliser les mêmes protocoles d'interaction pour tous les types.

DONNÉES DE TEXTES MISES EN FORME

Une fois atteinte la partie "données" (niveau d'exemple) dans le processus d'interrogation, le système nous présente un modèle du format des données. Un exemple de ce modèle est illustré à la figure 6.

Nous pouvons choisir des données dans un modèle semblable en utilisant les modes de spécification **remplissage** et (ou) **affichage** examinés précédemment. La valeur par défaut du remplissage de la zone suppose que nous voulons une correspondance parfaite pour la valeur précisée. Pour certains types de données, nous pouvons aussi prévoir d'autres sortes de correspondances: assortiments de structure (c'est-à-dire trouver une certaine structure dans une zone), inférieur à, inférieur ou égal à, supérieur à, supérieur ou égal à ainsi que concordances et discordances d'égalité. On peut indiquer

ces options en faisant précéder la valeur spécifiée d'un symbole approprié comme , , etc. Les zones qui doivent être remplies peuvent être choisies au moyen de l'opération de pointage examinée précédemment.

DONNÉES DE TEXTES NON MISES EN FORME

Si la partie données se compose de données de textes non mises en forme, nous obtenons alors la partie "données" qui nous est encore une fois présentée avec un modèle des données de textes non mises en forme (voir l'exemple à la figure 7).

Nous pouvons choisir des données dans un texte en précisant la structure recherchée, c'est-à-dire en indiquant dans la zone appropriée la structure du texte. À l'affichage de la structure spécifiée, le système nous montre la structure en contexte, c'est-à-dire l'endroit où elle apparaît dans le texte. Dans ce cas, les options **entrée suivante** et **entrée précédente** de l'opération **entrée** nous indiqueront les entrées suivantes et précédentes avec la structure en contexte (si la zone courante est constituée par le modèle). Nous pouvons aussi observer d'autres images avoisinantes au moyen de l'opération **image**. Tout le texte sera affiché si l'on passe en mode **affichage** sans remplir une formule.

SERVICE D'INFORMATION VIDÉOTEX

ITINÉRAIRE: Livres. Roman. Autant en emporte le vent. Chapitre 1

MESSAGE _____

Figure 7: Exemple de données non mises en forme

DONNÉES D'IMAGES ET DONNÉES VOCALES

Certaines zones de données de textes mises en forme ou non mises en forme peuvent correspondre à des données d'images ou à des données vocales. Actuellement, il est très difficile d'interroger ces types de données directement. À l'avenir, on pourra peut-être introduire comme données d'entrée une image ou un message vocal et demander au système tous les cas où on les retrouve. Cependant, pour l'instant, on doit se contenter d'afficher ou de reproduire une image ou les données vocales, en choisissant l'entrée appropriée d'une zone d'images ou d'une zone vocale au moyen de la fonction **choix**. Afin d'indiquer à l'utilisateur la nature de ces zones, on peut les faire ressortir au

Langages d'interrogation interactive pour bases de données externes

moyen de couleurs particulières ou leur apposer des symboles graphiques spéciaux, par exemple l'icône d'un appareil photo pour indiquer des photographies.

8 REMARQUES ET CONCLUSIONS

Nous avons examiné le processus de l'interrogation interactive en général, puis relativement à des bases de données externes en particulier. Le processus d'interrogation a été divisé en trois parties: la demande, la réponse et la dynamique. Pour chaque partie, nous avons identifié plusieurs caractéristiques et indiqué des propriétés souhaitables. Nous avons ensuite identifié sept langages d'interrogation et évalué chacun d'après ces caractéristiques. À partir de cette évaluation, nous avons établi certaines propriétés dont devraient être dotés les langages d'interrogation de bases de données externes. Finalement, nous avons proposé un spécimen auquel devrait correspondre ces langages d'interrogation.

Pendant l'élaboration de ce spécimen, nous avons fait porter nos efforts sur les aspects de requête et de dynamique du langage d'interrogation. Dans le cas des demandes, les paramètres de l'interaction utilisateur étaient les suivants: nombre de frappes requises, type et nombre de commandes disponibles, degré de liberté dans la formulation des demandes, sélectivité des demandes, uniformité des demandes par rapport aux bases de données consultées et aptitude à personnaliser la formulation de la demande relativement à une demande particulière. Dans les cas de la dynamique de l'interaction, les paramètres étaient les suivants: intensité entre l'utilisateur et le système, degré de ludicité de l'interaction, nombre de protocoles disponibles pour l'interaction, réaction du système à une demande de l'utilisateur et degré de contrôle de l'utilisateur dans une interaction. Voyons maintenant de quelle façon ces propriétés s'intègrent au spécimen proposé.

Le nombre de frappes requises pour préciser une demande dépend du mode de spécification choisi par l'utilisateur. Si celui-ci affiche et

marque des entrées, il lui suffit de pointer, marquer et choisir des valeurs. Le nombre de frappes nécessaires dépendra du pointeur disponible. Le marquage nécessite une frappe pour chaque valeur, tout comme la sélection. Si on a recours au mode remplissage, l'utilisateur peut alors pointer les zones et y faire entrer des valeurs. Cependant, même dans ce cas, de nombreuses frappes peuvent être évitées si l'utilisateur affiche les valeurs de la zone et les marque.

Pour le spécimen proposé, il y a deux modes de spécification, huit opérations de service et peut-être (selon le pointeur) de un à trois pointages. Un des deux modes de spécification sera le mode par défaut, que l'utilisateur n'aura pas besoin de préciser explicitement. Des huit opérations de service, il y a en deux dont on fera probablement un usage intensif: l'opération **choix** et l'opération **marquage**. Si des boutons portant le nom de la fonction correspondante sont prévus, l'utilisateur n'a pas à faire d'effort pour se souvenir du nom des fonctions. En outre, le nom des fonctions se veut représentatif et on peut toujours obtenir une aide en direct au moyen de la fonction **aide**.

En raison de la façon dont l'interaction utilisateur a été conçue, il est très difficile de formuler une demande incorrectement, que ce soit sur le plan de la syntaxe ou sur celui de la sémantique. Sur le plan de la syntaxe, le langage d'interrogation ne comprend pas de syntaxe dont il faut se souvenir. Les modes de spécification acceptent, pour les modèles, tout état d'entrée: vide, à moitié plein et plein à capacité. Les opérations de services sont toujours valides, mais elles peuvent donner lieu à un résultat nul. Sur le plan de la sémantique, les noms des zones apparaissent toujours dans le portique et l'utilisateur peut voir la structure des données au moyen du modèle. On peut donner des précisions inexactes sur une demande compliquée, c'est-à-dire s'il y a de nombreuses zones à qualifier au moyen de **et** et **ou** par exemple. Cependant, la façon dont les zones d'un modèle

interagissent par défaut (**et** entre les zones **ou** à l'intérieur des zones) est assez intuitive. On doit accepter que des demandes compliquées sont complexes et que leur spécification est plus facilement erronée.

La sélectivité du langage d'interrogation est presque aussi bonne que les mots-clés et les langages d'interrogation par exemple. Les sommes logiques sont les seules opérations difficiles à préciser (en fait nous n'avons pas dit de quelle façon les exécuter). Un des moyens consiste à utiliser une interaction conservée comme entrée d'une autre interaction. Cependant, nous ne croyons pas que la demande d'un langage d'interrogation de base de données externes permettant de résoudre des sommes logiques sera généralisée. Quoi qu'il en soit, on peut toujours avoir recours à une interface d'utilisateur plus expérimentée (comme un langage d'interrogation par mots-clés) pour ces demandes.

Dans le langage d'interrogation proposé, nous n'avons pas fait de supposition quant à la structure des bases de données externes. La nature de l'interface choisie - modèles - ne présuppose pas de structure de base de données sous-jacente particulière. Ainsi, on peut utiliser le langage d'interrogation pour assurer l'interface avec toute base de données, pourvu que des algorithmes appropriés soient prévus pour traduire les opérations et les structures du langage d'interrogation en opérations et structures du SGBD (système de gestion de base de données) de la base de données externe. On fait actuellement de nombreuses recherches dans ce secteur et plusieurs approches de la question peuvent être prises en considération (Date, 1980; Vassiliou and Lochovsky, 1980).

Le langage d'interrogation peut facilement s'adapter à des applications particulières. Il suffit, pour ce faire, de personnaliser les modèles dont se sert l'utilisateur pour interagir avec le système. Aucun autre aspect du langage d'interrogation n'a besoin d'être modifié. Plus particulièrement, le matériel requis pour l'interaction et les

Langages d'interrogation interactive pour bases de données externes

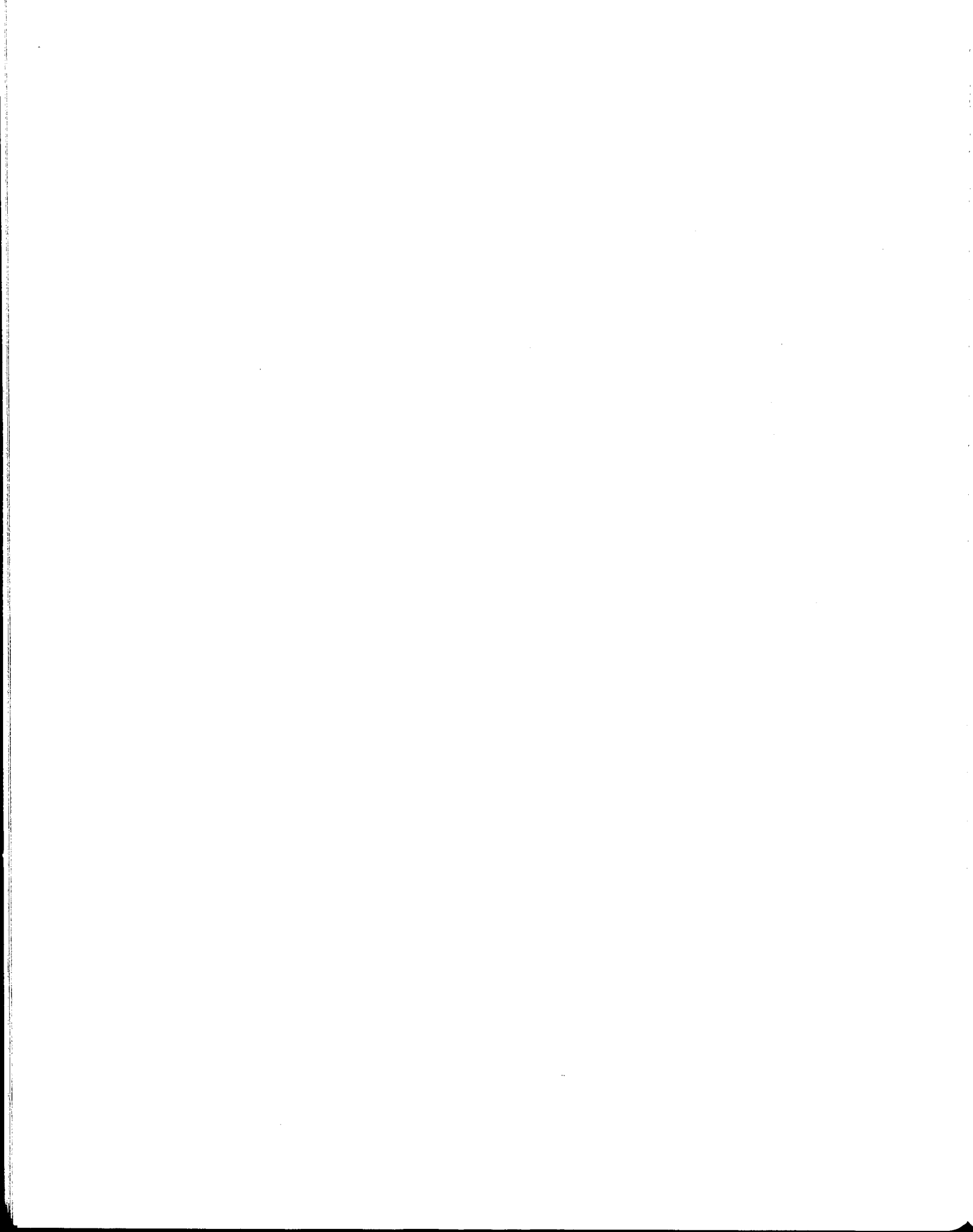
opérations de pointage, les opérations de services et les modes de spécification demeurent les mêmes.

L'intensité de l'interaction dépend du mode de spécification choisi. Pour l'affichage et le marquage, on peut avoir une interaction très intense. Pour le remplissage des modèles, elle peut être modérée. Il est à noter que les concepts sur lesquels repose l'interaction demeurent les mêmes lorsque nous passons du pointage au moyen d'un clavier à celui utilisant un dispositif mécanique, puis à un pointage au moyen d'instructions vocales. Ainsi, le langage d'interrogation peut tirer profit des progrès technologiques qui permettent d'accroître l'intensité de l'interaction.

La ludicité du langage d'interrogation varie de modérée à bonne. L'utilisateur peut explorer facilement d'autres trajets de la structure sans grand effort étant donné qu'il peut facilement conserver les trajets et(ou) rebrousser chemin en tout temps. Ce processus d'exploration permet à l'utilisateur de s'approcher graduellement de son but et de voir le chemin parcouru au moyen de la zone ITINÉRAIRE du portique. Selon la façon dont sont construits les renvois, l'utilisateur peut accéder aux mêmes données à partir de trajets différents. Le pointage, le marquage, l'affichage et le remplissage sont des types d'opération où l'utilisateur participe à l'interaction. L'utilisateur n'a pas seulement à appuyer sur un bouton. La conception des modèles et l'utilisation de couleurs et de sons peut ajouter à la ludicité du langage d'interrogation.

Il existe plusieurs protocoles permettant d'interroger les bases de données: le remplissage, l'affichage et les itinéraires préétablis. L'utilisateur peut choisir celui qui lui semble le mieux adapté à l'interaction donnée et à son niveau de compétence. Il peut aussi opter pour le mode de spécification par défaut selon la connaissance qu'il a du système.

Compte tenu de la technologie actuelle, la réponse du système peut être un facteur limitant le langage d'interrogation. Plus particulièrement, en mode **affichage**, le système doit afficher rapidement l'information demandée. Sur le plan des données, cela peut signifier l'extraction d'un grand nombre de renseignements. Pour qu'il y ait affichage des valeurs d'une zone quelconque, on doit disposer d'un mécanisme d'accès rapide pour extraire promptement toutes les valeurs. Au début, il faudra peut-être restreindre l'utilisation du mode **affichage** à la structure et à certaines zones des données pour surmonter les problèmes de réponse. Cependant, il se pourrait bien que les progrès technologiques viennent bientôt abolir cette restriction. Finalement, toute la nature de l'interaction de l'utilisateur - commander des affichages au système, marquer, etc. - contribue à donner à l'utilisateur le sentiment qu'il maîtrise la situation. C'est lui qui dirige le système pour son compte et pour atteindre son but. De plus, le paradigme d'interrogation que nous avons examiné précédemment contribue à créer ce sentiment. Il donne à l'utilisateur l'impression que le système est son allié puisqu'il l'aide à guider son vaisseau spatial (le système) dans un milieu hostile (l'espace informationnel) et qu'il le protège contre les dangers qui le guettent.



BIBLIOGRAPHIE

- Bonczek, R.H., Cash, J., et Whinston, A. (1977). "A Transformational Grammar-based Query Processor for Access Control in a Planning System," ACM TODS 2, pp. 326-338.
- Bonczek, E.H., et Whinston, A. (1977). "A Generalized Mapping Language for Network Data Structures," Information Systems 2, pp. 171-185.
- Bown, H.G., O'Brien, C.D., Sawchuk, W., et Storey, J.R. (1978). Description générale de Télidon: Proposition canadienne concernant les systèmes vidéotex. Note technique no 697-F. Centre de recherches sur les communications. Ministère des Communications, Ottawa.
- Bown, H., O'Brien, C.D., Sawchuk, W., Storey, J.R., et Treurniet, W.C. (1979). "Telidon Videotex and User-related Issues," Proc. Conf. on Visible Languages.
- Burger, J.F. (1977). "Data Base Semantics in the EUFID System," Proc. 2nd Berkeley Workshop on Distributed Data Management and Computer Networks, pp. 202-214.
- Carlson, C.R., et Kaplan, R.S. (1976). "A Generalized Access Path Model and its Application to Relational Data Base Systems," Proc. ACM SIGMOD, pp. 143-154.
- Chamberlin, D.D., et al. (1976). "SEQUEL2: A Unified Approach to Data Definition, Manipulation and Control," IBM Journal of Research and Development 20, pp. 560-575.
- Codd, E.F., Arnold, R.S., Cadiou, J-M., Chang, C.L., et Roussopoulos, N. (1978). RENDEZVOUS Version 1: An Experimental English-language Query Formulation System for Casual Users of Relational Data Bases Tech. rep. RJ2144, IBM Research Lab., San Jose, Calif.
- Date, C.J. (1976). "An Architecture for High-level Language Database Extensions," Proc. ACM SIGMOD, pp. 101-122.
- Date, C.J. (1980). "An Introduction to the Unified Database Language," Proc. ACM Intl. Conf. Very Large Data Bases, pp. 15-32.

- Deheneffe, C. et Hennebert, H. (1976). "NUL: A Navigational User's Language for a Network Structured Data Base," Proc. ACM SIGMOD, pp. 135-142.
- Denny, G.H. (1977). An Introduction to SQL, A Structured Query Language, Tech. Rep. RA93, IBM Research Lab., San Jose, Calif.
- Ellis, C.A., et Nutt, G.J. (1980). "Office Information Systems and Computer Science," ACM Computing Surveys 12, pp. 27-60.
- Gilb, T. et Weinberg, G.M. (1977). Humanized Input: Techniques for Reliable Keyed Input. Winthrop Publishers.
- Geudj, R.A., tenHagen, P.J.W., Hopgood, F.R.A., Tucker, H.A., et Duce, D.A. (eds) (1980). Methodology of Interaction, IFIP Workshop on Methodology of Interaction, Seillac, France. North-Holland, Amsterdam.
- Hammer, M., Howe, W.G., Kruskal, V.J., et Wladawsky, I (1977). "A Very High Level Programming Language for Data Processing Applications" CACM 20, pp. 832-840.
- Harris, L.R. (1977) "User Oriented Data Base Query with the ROBOT Natural Language Query System," Proc. ACM Intl. Conf. Very Large Data Bases, pp. 303-311.
- Hendrix, G.G. Sacerdoti, E.D., Sagalowicz, D., et Slocum, J. (1978). "Developing a Natural Language Interface to Complex Data," ACM TODS 3, pp. 105-147.
- Herot, C.F. (1980). "Spatial Management of Data," ACM TODS 5, pp. 493-513.
- Hogg, J. (1981). TLA: A System for Automating Form Procedures, M.Sc. thesis, Dept. of Computer Science, Univ. of Toronto.
- Hogg, J., Nierstrasz, O.M., et Tschritzis, D.C. (1981). "Form Procedures," in Omega Alpha, Tschritzis, D.C. (ed.), Tech. rep. CSRG-127, Computer Systems Research Group, Univ. of Toronto.
- Kerschberg, L., Ozkarahan, E.A., et Pacheco, J.E.S. (1976). "A Synthetic English Query Language for a Relational Associative Processor," Proc. 2nd Intl. Conf. on Software Engineering, pp. 505-519.
- Lacroix, M., et Pirotte, A. (1977a). ILL: An English Structured Query Language for Relational Data Bases," dans Architecture and Models in Data Base Management Systems, (Nijssen, G.M., ed.), pp. 237-260, North-Holland, Amsterdam.

Langages d'interrogation interactive pour bases de données externes

- Lacroix, M., et Pirotte, A. (1977b). "Domain-oriented Relational Languages," Proc. Intl. Conf. Very Large Data Bases, pp. 370-378.
- Lipson, W., et Lapczak, O. (1976). LSL User's Manual, Tech. note 9, Computer Systems Research Group, Univ. of Toronto.
- Luo, D., et Yao, S.B. (1981). "Form Operations By Example - A Language for Office Information Processing," Proc. ACM SIGMOD.
- McDonald, N., et Stonebraker, M.R. (1975). "CUPID - The Friendly Query Language," Proc. ACM Pacific 75 Regional Conf., pp. 127-131.
- Mantei, M. (1981). Disorientation Behaviour in Person-Computer Interaction, Ph.D. thesis, Dept. of Communications, Univ. of Southern California (à paraître).
- Martin, J. (1973). Design of Man-Computer Dialogues. Prentice-Hall, Inc., Englewood Cliffs, N.J.
- Mehlmann, M. (1981). When People Use Computers - An Approach to Developing an Interface. Prentice-Hall, Inc., Englewood Cliffs, N.J.
- Moorhead, W.G. (1976). GXRAM - A Relational Data Base Interface for Graphics, Tech. rep. RJ1735, IBM Research Lab., San Jose, Calif.
- Mylopoulos, J., Borgida, A., Cohen, P., Roussopoulos, N., Tsotsos, J., et Wong, H. (1975). "TORUS - Natural Language Understanding System for Data Management," Proc. 4th Intl. Joint Conf. Artificial Intelligence, pp. 414-421.
- Mylopoulos, J., Borgida, A., Cohen, P., Roussopoulos, N., Tsotsos, J., et Wong, H. (1976). "TORUS: A Step toward Bridging the Gap between Data Bases and the Casual User," Information Systems 2, pp. 49-64.
- Nierstrasz, O.M. (1981). Automatic Coordination and Processing of Electronic Forms in TLA (An Intelligent Office Information System), M.Sc. thesis, Dept. of Computer Science, Univ. of Toronto.
- Prenner, C.J., and Rowe, L.A. (1977). A Data Base Oriented Programming Language, ERL Memo, Electronic Research Laboratory, Univ. Calif. Berkeley.
- Reisner, P. (1977). "Use of Psychological Experimentation as an Aid to Development of a Query Language," IEEE Trans. on Software Engineering 3, pp. 218-229.

Langages d'interrogation interactive pour bases de données externes

- Reisner, P. (1981). "Human Factors Studies of Database Query Languages: A Survey and Assessment," ACM Computing Surveys 13.
- Reisner, P., Boyce, R.F., et Chamberlin, D.D. (1975). "Human Factors Evaluation of Two Data Base Query Languages - SQUARE and SEQUEL," Proc. AFIPS 44, NCC, pp. 447-452.
- Sagalowicz, D. (1977). "IDA: An Intelligent Data Access Program," Proc. ACM Intl. Conf. Very Large Data Bases, pp. 293-302.
- Schmidt, J.W. (1977). "Some High Level Language Constructs for Data of Type Relation," ACM TODS 2, pp. 247-261.
- Sharman, G.C.H. (1977). "Update-by-dialogue: An Interactive Approach to Data Base Manipulation," Proc. ACM SIGMOD, pp. 21-29.
- Shneiderman, B. (1978). "Improving the Human Factors Aspect of Database Interactions" ACM TODS 3, pp. 417-439.
- Shneiderman, B. (1980). Software Psychology: Human Factors in Computer and Information Systems. Prentice-Hall, Inc., Englewood Cliffs, N.J.
- Spath, C.R. et Schneider, L.S. (1977). "A Generalized End-user Facility Architecture for Relational Data Base Systems," Proc. ACM Intl. Conf. Very Large Data Bases, pp. 359-369.
- Stonebraker, M.R., et Rowe, L.A. (1977). "Observations on Data Manipulation Languages and Their Embedding in General Purpose Programming Languages," Proc. ACM Intl. Conf. Very Large Data Bases, pp. 128-143.
- Thomas, J.C. (1977). "Psychological Issues in Data Base Management," Proc. ACM Intl. Conf. Very Large Data Bases, pp. 169-185.
- Thomas, J.C., et Gould, J.D. (1975). "A Psychological Study of Query-by-example," Proc. AFIPS 44, NCC, pp. 439-445.
- Vandijck, E. (1977). "Towards a More Familiar Relational Retrieval Language," Information Systems 2, pp. 159-169
- Vassiliou, Y., et Lochovsky, F.H. (1980). "DBMS Transaction Translation," Proc. COMPSAC'80, pp. 89-96.
- Woods, W.A. (1973). "Progress in Natural Language Understanding - An Application to Lunar Geology," Proc. AFIPS 42, NCC, pp. 441-450.

Langages d'interrogation interactive pour bases de données externes

- Zloof, M.M. (1975a). "Query-by-example: The Invocation and Definition of Tables and Forms," Proc. ACM Intl. Conf. Very Large Data Bases, pp. 1-24.
- Zloof, M.M. (1975b). "Query-by-example," Proc. AFIPS 44, NCC, pp. 431-438.
- Zloof, M.M. (1977). "Query-by-example: A Data Base Language," IBM Systems Journal 16, pp. 324-343.
- Zloof, M.M. (1980). A Language for Office and Business Automation, IBM Tech. rep. RC8091, Yorktown Heights, New York.

Canada



Gouvernement du Canada
Ministère des Communications

Government of Canada
Department of Communications