

LES LANGAGES DE GESTION ET L'IMPLANTATION RATIONNELLE DES FICHIERS

par Jacques F. VALLÉE (1)

SOMMAIRE

Résumé. — Cet article explore les relations entre les problèmes de gestion et la documentation automatique. Il insiste sur la différenciation des notions de langage et de système et s'adresse plus particulièrement à la simplification des opérations dans l'établissement de fichiers de grande dimension, leur consultation, leur mise à jour et leur édition. Un langage expérimental appelé INFOL est discuté à la lumière des problèmes rencontrés par un groupe d'utilisateurs ; les progrès réalisés dans l'application de ce langage sont passés en revue. Une perspective est ainsi définie dans laquelle les performances souhaitables des futurs langages de gestion peuvent s'inscrire.

INTRODUCTION

Les Entreprises qui ont fait l'acquisition d'ensembles électroniques de troisième génération se trouvent actuellement placées devant de nouveaux problèmes de software qui forment trois groupes principaux :

- 1) Les carences typiques de ces systèmes en matière de communication à distance en temps partagé.
- 2) L'insuffisance des langages de programmation qui ne permettent pas de tirer le meilleur parti des vitesses d'accès et du volume des mémoires dans le traitement de fichiers de grande dimension.
- 3) Les limitations draconiennes que les langages classiques (aussi bien que les systèmes expérimentaux nouvellement mis en service) placent sur l'interface homme-machine.

(1) System Analyst, Vogelback Computing Center, Northwestern University, Evanston, Illinois, U.S.A.

D'une part, on déplore que les systèmes spécialisés ne permettent d'extraire un sous-ensemble d'un fichier que sur la base de critères restreints, sans possibilités de créer des variables secondaires ; les opérations que l'on peut effectuer ensuite sur ce sous-ensemble sont de toutes façons limitées. D'autre part, les utilisateurs ont reconnu que les langages classiques (COBOL, FORTRAN) ne permettaient pas de décrire les fichiers de gestion d'une manière suffisamment générale quand ils comportent des informations non-numériques à structure complexe.

Le présent article passe en revue certaines expériences récentes qui traitent des langages de gestion sous l'angle de la documentation automatique et fait le point des performances.

SYSTEMES ET LANGAGES

Dans un récent article portant sur le traitement de fichiers médicaux, Lamson et Dimsdale (5) ont caractérisé comme idéal un système qui « permettrait le stockage de documents et leur extraction de telle manière que l'utilisateur pourrait être inconscient de l'existence de l'ordinateur en opération ».

Un tel idéal, comme ils le remarquaient, est inaccessible dans le présent, mais les obstacles à sa réalisation peuvent être clairement définis. A notre avis, la confusion persistante entre la notion de « système » et celle de « langage » contribue à masquer certains problèmes : on lui doit en particulier la prolifération des programmes spécialisés, restreints à un ordinateur bien déterminé, parfois même à une configuration spéciale. La vulnérabilité de ces programmes aux changements d'environnement est connue. De tels « systèmes », conçus pour résoudre un problème précis et immédiat, semblent économiques à court terme parce qu'ils sont plus simples à écrire qu'un processeur général et ne demandent de la part du programmeur aucune connaissance théorique particulière. Au moment du passage en machine un système reçoit des ordres sous forme de cartes-paramètres dont le format est fixe. L'utilisateur doit être entraîné à l'emploi de ce format et du code des cartes. Il n'est à aucun moment « inconscient de l'existence de l'ordinateur en opération ».

Le développement de langages de gestion adaptés à l'implantation des fichiers représente un effort pour libérer l'utilisateur des contraintes que nous venons de rappeler, en chargeant la machine elle-même d'interpréter les requêtes en termes de codes, quel que soit le contexte de l'application considérée. La seule contribution qu'un langage de gestion demande de la part de l'utilisateur est la formalisation de la structure de son fichier et la description sans ambiguïté de sa terminologie. Nous nous proposons de décrire ici un tel langage, puis d'en analyser brièvement les applications pour tirer enfin certaines conclusions quant à l'évolution des ensembles réalisables pratiquement dans les années à venir.

D'une part, on déplore que les systèmes spécialisés ne permettent d'extraire un sous-ensemble d'un fichier que sur la base de critères restreints, sans possibilités de créer des variables secondaires ; les opérations que l'on peut effectuer ensuite sur ce sous-ensemble sont de toutes façons limitées. D'autre part, les utilisateurs ont reconnu que les langages classiques (COBOL, FORTRAN) ne permettaient pas de décrire les fichiers de gestion d'une manière suffisamment générale quand ils comportent des informations non-numériques à structure complexe.

Le présent article passe en revue certaines expériences récentes qui traitent des langages de gestion sous l'angle de la documentation automatique et fait le point des performances.

SYSTEMES ET LANGAGES

Dans un récent article portant sur le traitement de fichiers médicaux Lamson et Dimsdale (5) ont caractérisé comme idéal un système qui « permettrait le stockage de documents et leur extraction de telle manière que l'utilisateur pourrait être inconscient de l'existence de l'ordinateur en opération ».

Un tel idéal, comme ils le remarquaient, est inaccessible dans le présent, mais les obstacles à sa réalisation peuvent être clairement définis. A notre avis, la confusion persistante entre la notion de « système » et celle de « langage » contribue à masquer certains problèmes : on lui doit en particulier la prolifération des programmes spécialisés, restreints à un ordinateur bien déterminé, parfois même à une configuration spéciale. La vulnérabilité de ces programmes aux changements d'environnement est connue. De tels « systèmes », conçus pour résoudre un problème précis et immédiat, semblent économiques à court terme parce qu'ils sont plus simples à écrire qu'un processeur général et ne demandent de la part du programmeur aucune connaissance théorique particulière. Au moment du passage en machine un système reçoit des ordres sous forme de cartes-paramètres dont le format est fixe. L'utilisateur doit être entraîné à l'emploi de ce format et du code des cartes. Il n'est à aucun moment « inconscient de l'existence de l'ordinateur en opération ».

Le développement de langages de gestion adaptés à l'implantation des fichiers représente un effort pour libérer l'utilisateur des contraintes que nous venons de rappeler, en chargeant la machine elle-même d'interpréter les requêtes en termes de codes, quel que soit le contexte de l'application considérée. La seule contribution qu'un langage de gestion demande de la part de l'utilisateur est la formalisation de la structure de son fichier et la description sans ambiguïté de sa terminologie. Nous nous proposons de décrire ici un tel langage, puis d'en analyser brièvement les applications pour tirer enfin certaines conclusions quant à l'évolution des ensembles réalisables pratiquement dans les années à venir.

LES LANGAGES DE GESTION ET IMPLANTATION RATIONNELLE DES FICHIERS

par Jacques F. VALLÉE (1)

SOMMAIRE

Résumé. — *Cet article explore les relations entre les problèmes de gestion et la documentation automatique. Il insiste sur la différenciation des notions de langage et de système et s'adresse plus particulièrement à la simplification des opérations dans l'établissement de fichiers de grande dimension, leur consultation, leur mise à jour et leur édition. Le langage expérimental appelé INFOL est discuté à la lumière des problèmes rencontrés par un groupe d'utilisateurs ; les progrès réalisés dans l'application de ce langage sont présentés en revue. Une perspective est ainsi définie dans laquelle les performances souhaitées des futurs langages de gestion peuvent s'inscrire.*

INTRODUCTION

Les Entreprises qui ont fait l'acquisition d'ensembles électroniques troisième génération se trouvent actuellement placées devant de nouveaux problèmes de software qui forment trois groupes principaux :

- 1) Les carences typiques de ces systèmes en matière de communication à distance en temps partagé.
- 2) L'insuffisance des langages de programmation qui ne permettent pas de tirer le meilleur parti des vitesses d'accès et du volume des mémoires dans le traitement de fichiers de grande dimension.
- 3) Les limitations draconiennes que les langages classiques (aussi bien que les systèmes expérimentaux nouvellement mis en service) imposent sur l'interface homme-machine.

(1) System Analyst, Vogelback Computing Center, Northwestern University, Evanston, Illinois, U.S.A.

PRESENTATION SIMPLIFIEE DU LANGAGE INFOL

Le langage qui nous intéresse dans la suite de l'article a été introduit par la compagnie *Control Data* sous le nom d'INFOL. Nous l'avons adopté comme base en vue d'applications plus étendues que celles prévues à l'origine par les créateurs du langage. Le processeur que nous avons écrit comprend donc les fonctions essentielles d'INFOL et suit les règles du Manuel [1] pour la déclaration de la structure des données et la création des fichiers. Au moment de l'interrogation, notre version déborde du cadre d'origine par plusieurs fonctions nouvelles décrites ci-dessous. Nous désignerons ce nouveau processeur sous le nom d'INFOL 2 pour le différencier du premier.

Control Data distribua INFOL en 1966 pour utilisation exclusive sur ses modèles 3600 et 3800. Le processeur lui-même se compose de 183 sous-programmes qui forment six overlays et est écrit en langage symbolique (COMPASS). Il passe sur une machine dotée d'une mémoire de 32 K.

Le milieu de stockage est la bande magnétique, avec une configuration variable selon le type du travail.

Les auteurs de ce premier processeur (une équipe dirigée par T. W. Olle) ont introduit certaines définitions intéressantes pour un fichier hypothétique général, dont les fichiers à traiter seront tous des cas particuliers. Il est composé d'*éléments* qui consistent en listes d'*items*. Ceux-ci peuvent être uniques (par exemple le nom d'un malade dans le fichier d'un hôpital) ou *multiples* (comme la liste des diagnostics, les opérations subies, etc.). Les expressions comme NOM DU MALADE, DIAGNOSTICS ou OPERATIONS deviennent les *descriptions des items*. Elles font partie des informations déclarées une fois pour toutes par l'utilisateur qui crée un fichier. Les données sont traitées en longueur variable, ce qui permet de perforer les informations de chaque fiche sans adhérer à un format déterminé.

Le langage opère en six *phases* distinctes.

1. Création (Establishment)

L'information déclarative est lue, les données composant le fichier sont traitées avec éventuellement les conversions de type qui sont nécessaires. Une bande finale est préparée.

2. Validation

Si les déclarations sont trouvées incompatibles par rapport aux données, des messages d'erreur sont imprimés.

3. Comptabilité (Bookkeeping)

Le processeur tient à jour une table de la dimension maximale des informations lues pour chaque item, soit en forme directe, soit en forme effective après conversion. Cette table est imprimée en clair en tête du rapport final.

4. Interrogations

Les critères fournis par l'utilisateur déterminent l'extraction d'éléments formant un sous-fichier. Pour ces éléments, certains items sont imprimés dans le rapport final.

5. Mise à jour (Update)

La modification du fichier peut être *discrète* quand elle s'applique à un élément particulier dont l'indicatif est fourni à l'ordinateur, ou *sélective* quand elle affecte tous les éléments dont les items vérifient des critères d'extraction.

6. Révision

Cette phase ne change pas les valeurs inscrites dans les fichiers, mais la structure de l'ensemble. Ainsi on peut ajouter de nouveaux items, changer les descriptions, les types et catégories, tout en évitant une nouvelle perforation très coûteuse de la masse des données.

Dans la première version expérimentale d'INFOL 2 qui s'adressait à la structure du langage plutôt qu'à son exploitation massive, les phases 5 et 6 n'ont pas été implantées. (La programmation de ces phases est actuellement en cours.) Par contre, les phases 2 et 3 ont été incorporées au programme principal et ne seront plus reconnues comme phases autonomes. Elles sont devenues des fonctions automatiques du processeur, de telle sorte qu'un contrôle absolu est maintenu en permanence sur le fichier.

EXEMPLE D'APPLICATION

Supposons maintenant que l'on utilise INFOL pour implanter le fichier Personnel d'une organisation. Il est naturel de reconnaître un employé individuel comme l'élément de base caractérisé par un certain nombre d'items. On déclare pour chaque item une *catégorie* (Unique ou Multiple) et un *type* : *Numérique* s'il s'agit d'un nombre entier ou réel, *date* pour permettre l'utilisation de n'importe quelle date usuelle, *code* s'il s'agit d'information codée sous forme mnémonique ou numérique, et enfin *alphanumérique* si l'on veut pouvoir lire un nom, une expression, ou un texte (contenant n'importe quel caractère sauf un astérisque).

Plusieurs items multiples consécutifs peuvent être réunis en un groupe d'association s'ils ont toujours un nombre égal de sous-items. Enfin, l'utilisateur définira ses propres codes et un ensemble de critères de validation ; ainsi, il déclarera certains items comme *nécessaires* ou imposera un nombre *maximum* de sous-items. Il définira un *intervalle* pour certaines valeurs, ou même assujettira la forme des données : *Alphabétique*, *Entier*, *non numérique*. Un nombre *maximum de caractères* peut aussi être spécifié. Avec ces définitions à l'esprit nous pouvons suivre l'exemple de la figure 1 qui montre les déclarations nécessaires pour un fichier simple.

Pendant la phase d'interrogation, une recherche est effectuée sur la bande. Chaque élément est examiné et soumis à des *critères* conduisant à

4. Interrogations

Les critères fournis par l'utilisateur déterminent l'extraction d'éléments formant un sous-fichier. Pour ces éléments, certains items sont imprimés dans le rapport final.

5. Mise à jour (Update)

La modification du fichier peut être *discrète* quand elle s'applique à un élément particulier dont l'indicatif est fourni à l'ordinateur, ou *sélective* quand elle affecte tous les éléments dont les items vérifient des critères d'extraction.

6. Révision

Cette phase ne change pas les valeurs inscrites dans les fichiers, mais la structure de l'ensemble. Ainsi on peut ajouter de nouveaux items changer les descriptions, les types et catégories, tout en évitant une nouvelle perforation très coûteuse de la masse des données.

Dans la première version expérimentale d'INFOL 2 qui s'adressait à la structure du langage plutôt qu'à son exploitation massive, les phases 1 et 6 n'ont pas été implantées. (La programmation de ces phases est actuellement en cours.) Par contre, les phases 2 et 3 ont été incorporées au programme principal et ne seront plus reconnues comme phases autonomes. Elles sont devenues des fonctions automatiques du processeur de telle sorte qu'un contrôle absolu est maintenu en permanence sur le fichier.

EXEMPLE D'APPLICATION

Supposons maintenant que l'on utilise INFOL pour implanter le fichier Personnel d'une organisation. Il est naturel de reconnaître un employé individuel comme l'élément de base caractérisé par un certain nombre d'items. On déclare pour chaque item une *catégorie* (Unique ou Multiple) et un *type* : *Numérique* s'il s'agit d'un nombre entier ou réel *date* pour permettre l'utilisation de n'importe quelle date usuelle, *codée* s'il s'agit d'information codée sous forme mnémonique ou numérique, et enfin *alphanumérique* si l'on veut pouvoir lire un nom, une expression ou un texte (contenant n'importe quel caractère sauf un astérisque).

Plusieurs items multiples consécutifs peuvent être réunis en un groupe d'association s'ils ont toujours un nombre égal de sous-items. Enfin, l'utilisateur définira ses propres codes et un ensemble de critères de validation ; ainsi, il déclarera certains items comme *nécessaires* ou imposera un nombre *maximum* de sous-items. Il définira un *intervalle* pour certaines valeurs, ou même assujettira la forme des données : *Alphabétique*, *Entier non numérique*. Un nombre *maximum de caractères* peut aussi être spécifié. Avec ces définitions à l'esprit nous pouvons suivre l'exemple de la figure 1 qui montre les déclarations nécessaires pour un fichier simple.

Pendant la phase d'interrogation, une recherche est effectuée sur la bande. Chaque élément est examiné et soumis à des *critères* conduisant à

PRESENTATION SIMPLIFIEE DU LANGAGE INFOL

Le langage qui nous intéresse dans la suite de l'article a été introduit par la compagnie *Control Data* sous le nom d'INFOL. Nous l'avons adopté comme base en vue d'applications plus étendues que celles prévues d'origine par les créateurs du langage. Le processeur que nous avons fait comprendre donc les fonctions essentielles d'INFOL et suit les règles du Manuel [1] pour la déclaration de la structure des données et la création des fichiers. Au moment de l'interrogation, notre version déborde du cadre d'origine par plusieurs fonctions nouvelles décrites ci-dessous. Nous désignerons ce nouveau processeur sous le nom d'INFOL 2 pour différencier du premier.

Control Data distribua INFOL en 1966 pour utilisation exclusive sur ses modèles 3600 et 3800. Le processeur lui-même se compose de 183 sous-programmes qui forment six overlays et est écrit en langage symbolique (COMPASS). Il passe sur une machine dotée d'une mémoire de 32 K.

Le milieu de stockage est la bande magnétique, avec une configuration variable selon le type du travail.

Les auteurs de ce premier processeur (une équipe dirigée par T. W. Olle) ont introduit certaines définitions intéressantes pour un fichier hypothétique général, dont les fichiers à traiter seront tous des cas particuliers. Il est composé d'*éléments* qui consistent en listes d'*items*. Ceux-ci peuvent être uniques (par exemple le nom d'un malade dans le fichier d'un hôpital) ou multiples (comme la liste des diagnostics, les opérations subies, etc.). Ces expressions comme NOM DU MALADE, DIAGNOSTICS ou OPERATIONS deviennent les *descriptions des items*. Elles font partie des informations déclarées une fois pour toutes par l'utilisateur qui crée un fichier. Les données sont traitées en longueur variable, ce qui permet de perforer les informations de chaque fiche sans adhérer à un format déterminé.

Le langage opère en six *phases* distinctes.

1. Création (Establishment)

L'information déclarative est lue, les données composant le fichier sont traitées avec éventuellement les conversions de type qui sont nécessaires. Une bande finale est préparée.

2. Validation

Si les déclarations sont trouvées incompatibles par rapport aux données, des messages d'erreur sont imprimés.

3. Comptabilité (Bookkeeping)

Le processeur tient à jour une table de la dimension maximale des informations lues pour chaque item, soit en forme directe, soit en forme effective après conversion. Cette table est imprimée en clair en tête du rapport final.

l'extraction possible des valeurs de certains items, comme le montrent les exemples suivants : supposons que l'on s'adresse à nouveau au fichier de la figure 1.

On veut d'abord, sous forme de rapport, la liste des noms de tous les employés dont la fiche ne comporte pas le statut marital. L'instruction suivante répond à la question :

RETRIEVAL CRITERIA 1 *9* DOES NOT EXIST
EXTRACTIONS 1 *2* REPORT

Si l'on veut le nom et le statut marital de tous les employés engagés avant le 1.12.66, on écrira :

RETRIEVAL CRITERIA 2 *3* LE 1.12.66
EXTRACTIONS 2 *2* REPORT *9* REPORT

Pour avoir tous les numéros des employés dont le statut militaire a été intégré au fichier en septembre 1965.

RETRIEVAL CRITERIA 3 *8* ENTRY DATE EQ SEPTEMBER
1965
EXTRACTIONS 3 *1* REPORT

Pour obtenir un rapport donnant les noms de tous les employés dont le salaire était inférieur à 1500 avant leur seconde révision et dont la dernière performance a été jugée « excellent » ou « très bien ».

RETRIEVAL CRITERIA 4 *4* SUB ITEM 2 LT 1500
6 LAST EQ (EX OR TB)
EXTRACTIONS 4 *2* REPORT

Dans les comparaisons de dates et de nombres, tous les opérateurs de relations usuels sont autorisés. Dans la désignation des sous-items et des quantités dérivées, on peut spécifier : FIRST, SUB-ITEM *n*, LAST BUT *n*, ANY, ALL, MAXIMUM, MINIMUM, SUM, MEAN, TOTAL. Ainsi, pour extraire les fiches des employés célibataires dont le salaire maximum a dépassé 3 000, on écrirait :

RETRIEVAL CRITERIA 5 *4* MAXIMUM GE 3 000
9 EQ 1
EXTRACTIONS 5 *1* REPORT

Pour obtenir en clair (décodée) la performance et le nom de tous les employés dont le statut n'a pas été révisé depuis le 12 février 1966 et qui sont entrés avant 1964, on demanderait :

RETRIEVAL CRITERIA 6 *7* ALL LE 12/FEB/66
3 LT 1964
EXTRACTIONS 6 *6* REPORT DECODED
2 REPORT

INFOLD
ESTABLISHMENT
ITEM DESCRIPTIONS 9
NUMERO DE FICHE *1* NOM *2* DATE D'ENGAGEMENT *3* SALAIRE MENSUEL *4*
PERIODES PAYEES *5* PERFORMANCE *6* DATE DE REVISION *7* STATUT MILITAIRE *8*
STATUT MARITAL *9*
CATEGORY-TYPE
1 UNARY NUMERIC
2 UNARY ALPHANUMERIC
3 UNARY DATE
4 MULTIPLE NUMERIC ASSOCIATION 3
5 MULTIPLE NUMERIC ASSOCIATION 3
6 MULTIPLE CODED ASSOCIATION 3
7 MULTIPLE DATE ASSOCIATION 3
8 UNARY ALPHANUMERIC
9 UNARY CODED
CODES
6 MNEMONIC EX EXCELLENT * TB TRÈS BIEN * B BIEN * SA SATISFAISANT *
IS INFÉRIEUR AUX STANDARDS * TM TRÈS MAUVAIS *
9 NUMBER CELIBATAIRE * DIVORCE * VEUVE AVEC ENFANT * MARIÉ SANS ENFANT *
MARIÉ AVEC ENFANT **
VALIDATIONS
1 RANGE 1 9999 INTEGER *2* NECESSARY CHARACTERS 36 ALPHABETIC
3 RANGE 1/2/1965 TODAY *4* INTEGER *5* RANGE JAN 1965 TODAY
7 RANGE JAN 1965 TODAY
INITIAL INPUT
1 689 *2* ROBERT UNTEL *3* 2 JAN 1965 *4* 1000 * 1025 * 1200 *
5 JAN 1965 * JUNE 1965 * DEC 1965 * *6* SA * TB * EX *
7 JAN 1965 * JULY 1965 * JAN 1966 *
8 LIBERE DES OBLIGATIONS MILITAIRES (AVIATION 1960-61) *9* 1 *

Figure 1

Dans le premier langage INFOL, les valeurs des items sélectionnées étaient imprimées verticalement, ainsi que les descriptions des items. Pour obtenir un rapport plus lisible sous forme de table, l'utilisateur devait employer un vocabulaire spécial qui n'apparaît pas dans les exemples précédents. Dans INFOL 2, une grande partie de ces fonctions sont devenues automatiques.

EXPERIENCES D'APPLICATION ET D'EXTENSION DU LANGAGE

Une série d'expériences conduisant au nouveau processeur, a été faite en 1967. Cette nouvelle version du langage reconnaît strictement le vocabulaire d'entrée utilisé par INFOL et a même été mise au point sur les mêmes données que le processeur d'origine. Certaines différences mineures existent dans l'interprétation des déclarations : en particulier, un item numérique est considéré comme réel à moins d'être déclaré entier. Les déclarations de code ont été simplifiées dans le cas où des codes identiques sont utilisés par plusieurs items. Cette situation a été rencontrée dans le

```

INFOL 0
ESTABLISHMENT
ITEM DESCRIPTIONS 9
  NUMERO DE FILLE *1* NOM *2* DATE D'ENGAGEMENT *3* SALAIRE MENSUEL *4*
  PERIODES PAYEES *5* PERFORMANCE *6* DATE DE REVISION *7* STATUT MILITAIRE *8*
  STATUT MARITAL *9*
CATEGORY-TYPE
  *1* UNARY NUMERIC
  *2* UNARY ALPHANUMERIC
  *3* UNARY DATE
  *4* MULTIPLE NUMERIC ASSOCIATION 3
  *5* MULTIPLE NUMERIC ASSOCIATION 3
  *6* MULTIPLE CODED ASSOCIATION B
  *7* MULTIPLE DATE ASSOCIATION B
  *8* UNARY ALPHANUMERIC
  *9* UNARY CODED
CODES
  *0* MNEMONIC  EX EXCELLENT * TB TRES BIEN * B BIEN * SA SATISFAISANT *
  IS INFERIEUR AUX STANDARDS * TM TRES MAUVAIS *
  *9* NUMBER  CELIBATAIRE * DIVORCE * VEUF AVEC ENFANT * MARIE SANS ENFANT *
  MARIE AVEC ENFANT **
VALIDATIONS
  *1* RANGE 1 9999 INTEGER  *2* NECESSARY CHARACTERS 36 ALPHABETIC
  *3* RANGE 1/2/1965 TODAY  *4* INTEGER  *5* RANGE JAN 1965 TODAY
  *7* RANGE JAN 1965 TODAY
INITIAL INPUT
*1* 009  *2* ROBERT UNTIL  *3* 2 JAN 1965  *4* 1000 * 1025 * 1200 *
*5* JAN 1965 * JUNE 1965 * DEC 1965 *  *6* SA * TB * EX *
*7* JAN 1965 * JULY 1965 * JAN 1966 *
*8* LIBERE DES OBLIGATIONS MILITAIRES (AVIATION 1960-61)  *9* 1 *

```

Figure 1

Dans le premier langage INFOL, les valeurs des items sélectionnées étaient imprimées verticalement, ainsi que les descriptions des items. Pour obtenir un rapport plus lisible sous forme de table, l'utilisateur devait employer un vocabulaire spécial qui n'apparaît pas dans les exemples précédents. Dans INFOL 2, une grande partie de ces fonctions sont devenues automatiques.

EXPERIENCES D'APPLICATION ET D'EXTENSION DU LANGAGE

Une série d'expériences conduisant au nouveau processeur, a été faite en 1967. Cette nouvelle version du langage reconnaît strictement le vocabulaire d'entrée utilisé par INFOL et a même été mise au point sur les mêmes données que le processeur d'origine. Certaines différences mineures existent dans l'interprétation des déclarations : en particulier, un item numérique est considéré comme réel à moins d'être déclaré entier. Les déclarations de code ont été simplifiées dans le cas où des codes identiques sont utilisés par plusieurs items. Cette situation a été rencontrée dans le

Extraction possible des valeurs de certains items, comme le montrent les exemples suivants : supposons que l'on s'adresse à nouveau au fichier de la figure 1.

On veut d'abord, sous forme de rapport, la liste des noms de tous les employés dont la fiche ne comporte pas le statut marital. L'instruction suivante répond à la question :

RETRIEVAL CRITERIA 1 *9* DOES NOT EXIST
EXTRACTIONS 1 *2* REPORT

Si l'on veut le nom et le statut marital de tous les employés engagés avant le 1.12.66, on écrira :

RETRIEVAL CRITERIA 2 *3* LE 1.12.66
EXTRACTIONS 2 *2* REPORT *9* REPORT

Pour avoir tous les numéros des employés dont le statut militaire a été intégré au fichier en septembre 1965.

RETRIEVAL CRITERIA 3 *8* ENTRY DATE EQ SEPTEMBER
1965
EXTRACTIONS 3 *1* REPORT

Pour obtenir un rapport donnant les noms de tous les employés dont le salaire était inférieur à 1500 avant leur seconde révision et dont la dernière performance a été jugée « excellent » ou « très bien ».

RETRIEVAL CRITERIA 4 *4* SUB ITEM 2 LT 1500
6 LAST EQ (EX OR TB)
EXTRACTIONS 4 *2* REPORT

Dans les comparaisons de dates et de nombres, tous les opérateurs de relations usuels sont autorisés. Dans la désignation des sous-items et des quantités dérivées, on peut spécifier : FIRST, SUB-ITEM *n*, LAST, LT *n*, ANY, ALL, MAXIMUM, MINIMUM, SUM, MEAN, TOTAL. Ainsi, pour extraire les fiches des employés célibataires dont le salaire maximum a dépassé 3 000, on écrirait :

RETRIEVAL CRITERIA 5 *4* MAXIMUM GE 3 000
9 EQ 1
EXTRACTIONS 5 *1* REPORT

Pour obtenir en clair (décodée) la performance et le nom de tous les employés dont le statut n'a pas été révisé depuis le 12 février 1966 et qui ont été entrés avant 1964, on demanderait :

RETRIEVAL CRITERIA 6 *7* ALL LE 12/FEB/66
3 LT 1964
EXTRACTIONS 6 *6* REPORT DECODED
2 REPORT

cas des fichiers médicaux où des diagnostics codés dans le même système peuvent apparaître à plusieurs niveaux.

Dans ces expériences (faites principalement à l'Université North-western) nous avons imposé sur le fichier les limitations maximales de trois cents items par fiche, sept cents caractères par item et deux mille mots (16 000 caractères sur la CDC 3 400) pour l'ensemble d'un élément. Les éléments composant le fichier total sont inscrits sur bande en longueur variable et précédés d'un répertoire décrivant les caractéristiques des items, la structure du fichier et sa terminologie spécifique (fig. 2).

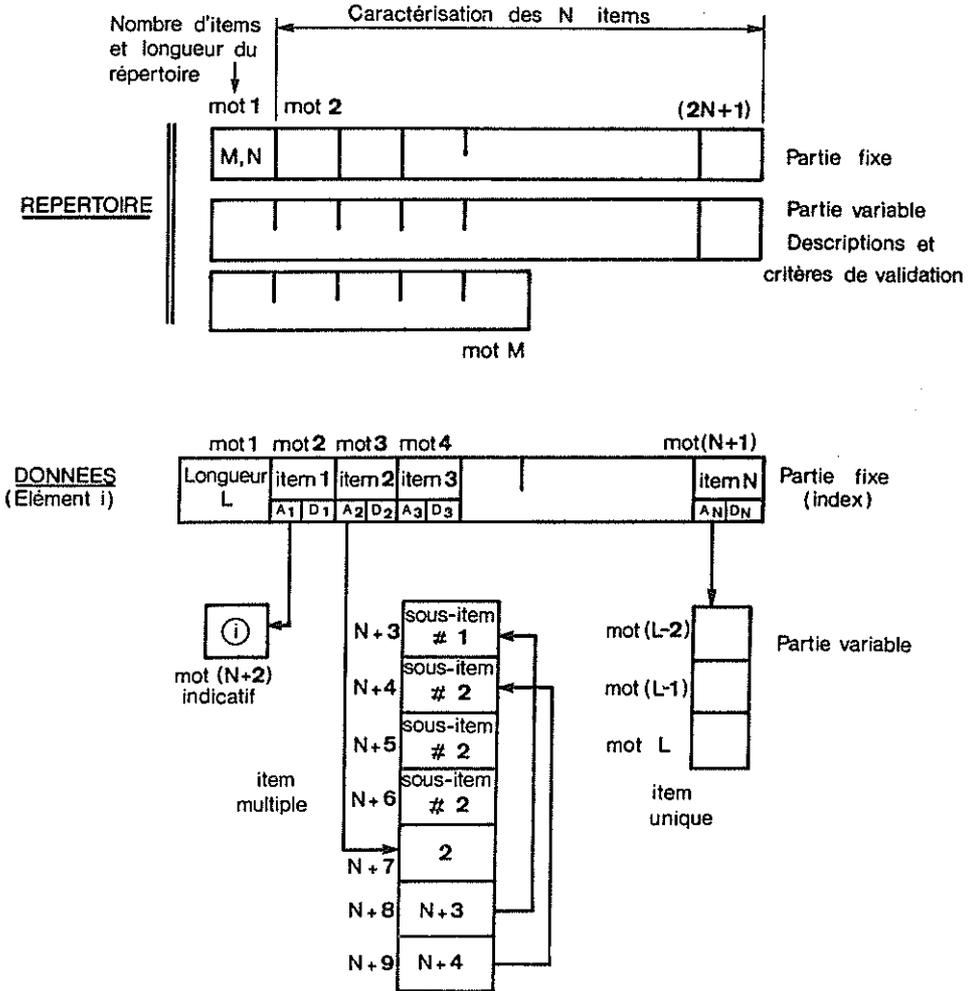


Figure 2
Structure du Fichier

La détection des erreurs conduit à un ensemble de diagnostics répondant à deux objectifs : 1) informer l'utilisateur le plus clairement possible sur la nature de son erreur, sa position et les moyens de la corriger et 2) éviter d'interrompre le traitement du fichier quand l'erreur n'entraîne pas d'incompatibilité interne. Par exemple, une date fautive sera détectée et clairement identifiée. Un code spécial sera écrit sur la bande et le rapport final reconnaîtra ce code à son tour pour le remplacer par l'expression BAD DATE.)

Certaines modifications opérationnelles ont été introduites pour permettre à INFOL 2 d'intégrer les différentes phases de traitement et en particulier de faire suivre la création d'un fichier par son interrogation sans changement de bandes ni ré-initialisation externe. La figure 3 montre l'organisation des principales fonctions pour la phase de *création*.

Des différences majeures existent entre INFOL et INFOL 2 au moment de l'*extraction* et de l'impression des rapports. Au cours des expériences traitées en collaboration avec l'Université Northwestern (2) nous avons sélectionné pour implantation des fichiers-types fournis par plusieurs départements et comprenant des mélanges variés d'informations numériques et non numériques : ainsi un catalogue astronomique nous a donné un environnement à dominance numérique. Le fichier cardiologique de l'Hôpital des Enfants-Malades de Chicago a été implanté par R. Chalice et l'auteur en utilisant INFOL 2. Cette application nous a fourni un exemple d'une chaîne d'items multiples comprenant des événements codés, des diagnostics alphanumériques et des dates. Un fichier chirurgical et un fichier dermatologique ont montré au contraire l'importance des résumés en langage naturel et de descriptions verbales (techniques opératoires, descriptions de lésions, etc.). Ces deux derniers fichiers comportaient une faible proportion d'items à valeurs numériques.

Dans ces applications qui constituaient à notre connaissance le premier test étendu du langage INFOL, nous avons trouvé que les règles définies à l'origine étaient trop étroites pour nous permettre d'utiliser toute la flexibilité que la structure générale du fichier autorisait, en particulier lorsque la multiplicité des informations devait être traitée à plusieurs niveaux. Nous avons été conduits à introduire dans le langage *trois fonctions nouvelles* :

1. La recherche de mots-clé dans les textes alphanumériques ou les suites de codes.
2. Le groupement logique de critères de sélection en unités de niveau plus élevé.
3. L'association automatique des sous-items homologues dans tous les items multiples d'un même fichier.

L'exemple suivant illustre les deux premières fonctions. Toujours avec le fichier de la figure 1, les critères de sélection suivants pourraient être utilisés pour extraire les employés qui

1. sont mariés sans enfant *ou* gagnent moins de 1 500 et

La détection des erreurs conduit à un ensemble de diagnostics répondant à deux objectifs : 1) informer l'utilisateur le plus clairement possible sur la nature de son erreur, sa position et les moyens de la corriger et 2) éviter d'interrompre le traitement du fichier quand l'erreur n'entraîne pas d'incompatibilité interne. Par exemple, une date fautive sera détectée et clairement identifiée. Un code spécial sera écrit sur la bande et le rapport final reconnaîtra ce code à son tour pour le remplacer par l'expression BAD DATE.)

Certaines modifications opérationnelles ont été introduites pour permettre à INFOL 2 d'intégrer les différentes phases de traitement et en particulier de faire suivre la création d'un fichier par son interrogation sans changement de bandes ni ré-initialisation externe. La figure 3 montre l'organisation des principales fonctions pour la phase de *création*.

Des différences majeures existent entre INFOL et INFOL 2 au moment de l'*extraction* et de l'impression des rapports. Au cours des expériences traitées en collaboration avec l'Université Northwestern (2) nous avons sélectionné pour implantation des fichiers-types fournis par plusieurs départements et comprenant des mélanges variés d'informations numériques et non numériques : ainsi un catalogue astronomique nous a donné un environnement à dominance numérique. Le fichier cardiologique de l'Hôpital des Enfants-Malades de Chicago a été implanté par R. Chalice et l'auteur en utilisant INFOL 2. Cette application nous a fourni un exemple d'une chaîne d'items multiples comprenant des événements codés, des diagnostics alphanumériques et des dates. Un fichier chirurgical et un fichier dermatologique ont montré au contraire l'importance des résumés en langage naturel et de descriptions verbales (techniques opératoires, descriptions de lésions, etc.). Ces deux derniers fichiers comportaient une faible proportion d'items à valeurs numériques.

Dans ces applications qui constituaient à notre connaissance le premier test étendu du langage INFOL, nous avons trouvé que les règles définies à l'origine étaient trop étroites pour nous permettre d'utiliser toute la flexibilité que la structure générale du fichier autorisait, en particulier lorsque la multiplicité des informations devait être traitée à plusieurs niveaux. Nous avons été conduits à introduire dans le langage *trois fonctions nouvelles* :

1. La recherche de mots-clé dans les textes alphanumériques ou les suites de codes.
2. Le groupement logique de critères de sélection en unités de niveau plus élevé.
3. L'association automatique des sous-items homologues dans tous les items multiples d'un même fichier.

L'exemple suivant illustre les deux premières fonctions. Toujours avec le fichier de la figure 1, les critères de sélection suivants pourraient être utilisés pour extraire les employés qui

1. sont mariés sans enfant *ou* gagnent moins de 1 500 *et*

s des fichiers médicaux où des diagnostics codés dans le même système peuvent apparaître à plusieurs niveaux.

Dans ces expériences (faites principalement à l'Université North-eastern) nous avons imposé sur le fichier les limitations maximales de trois cents items par fiche, sept cents caractères par item et deux mille mots (6 000 caractères sur la CDC 3 400) pour l'ensemble d'un élément. Les éléments composant le fichier total sont inscrits sur bande en longueur variable et précédés d'un répertoire décrivant les caractéristiques des items, la structure du fichier et sa terminologie spécifique (fig. 2).

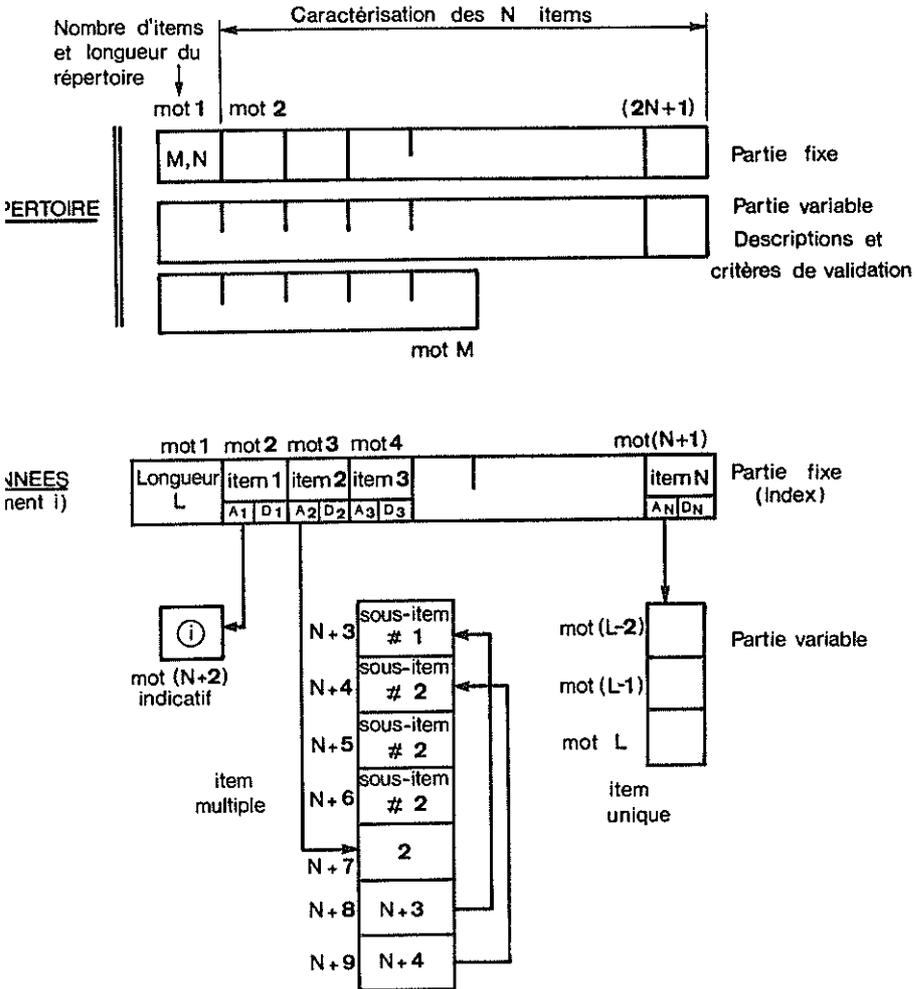


Figure 2
Structure du Fichier

2. sont libérés du service militaire mais n'ont pas été réformés ou exemptés :

RETRIEVAL CRITERIA 7 *9* EQ 4 OR *4* LAST LT 1 500
8 EQ LIBERE AND NE
(REFORME OR EXEMPTÉ)

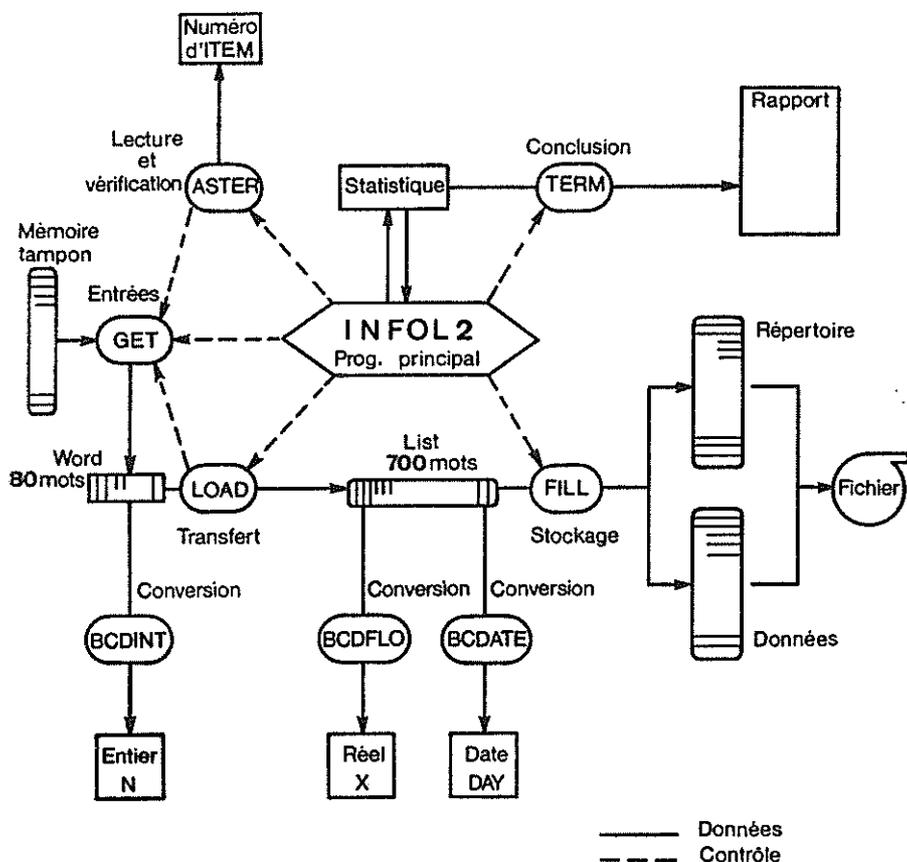


Figure 3

Création d'un fichier avec INFOL 2 : organisation simplifiée du processeur

(Le lecteur pourra vérifier que l'employé n° 689 pris comme exemple figure 1 serait sélectionné par ces critères.) Les deux instructions ci-dessus constituent l'ensemble du « programme » INFOL 2 que l'on doit écrire pour effectuer cette sélection.

La troisième fonction (association automatique) est complexe. Nous donnerons deux exemples de son application.

EXEMPLE 1

On donne un fichier de comptabilité comportant en particulier les destinations, les dates et le volume des livraisons pour un produit donné respectivement, les items multiples n° 15, 7 et 13. Ainsi la fiche du produit *X* pourrait être :

ITEM 7 Date	ITEM 13 Unités transportées	ITEM 15 Destination
5/MAY/1967	1 375	Genève
2/MAY/1967	655	Marseille
10/JAN/1967	2 900	Marseille
21/FEB/1967	700	—
7/APR/1967	1 825	Rome
3/MAR/1967	530	Milan
6/MAY/1967	1 650	Marseille
5/JUN/1967	—	Rotterdam
—	—	Marseille

Remarquer que ces items ne sont pas consécutifs et n'ont pas le même nombre de sous-items, certaines informations étant absentes. Ils ne pourraient constituer un « Groupe d'association » au sens du langage INFOL d'origine.

On demande de sélectionner les produits pour lesquels des livraisons supérieures à 1 500 unités ont été faites à Marseille dans la première semaine de mai 1967.

Le programme INFOL 2 qui résout ce problème est le suivant :

RETRIEVAL CRITERIA 1 *15* ANY EQ MARSEILLE

7 SUB ITEM ASSOCIATED GE

1/MAY 1967 AND LE

7/MAY/1967

13 SUB ITEM ASSOCIATED GT

1 500

EXEMPLE 2

On donne un fichier de répartition géographique des ventes de produits pétroliers pour la France, chaque fiche représentant un mois. Les données

La troisième fonction (association automatique) est complexe. Nous donnerons deux exemples de son application.

EXEMPLE 1

On donne un fichier de comptabilité comportant en particulier les destinations, les dates et le volume des livraisons pour un produit donné respectivement, les items multiples n° 15, 7 et 13. Ainsi la fiche du produit X pourrait être :

ITEM 7 Date	ITEM 13 Unités transportées	ITEM 15 Destination
5/MAY/1967	1 375	Genève
2/MAY/1967	655	Marseille
10/JAN/1967	2 900	Marseille
21/FEB/1967	700	—
7/APR/1967	1 825	Rome
3/MAR/1967	530	Milan
6/MAY/1967	1 650	Marseille
5/JUN/1967	—	Rotterdam
—	—	Marseille

Remarquer que ces items ne sont pas consécutifs et n'ont pas le même nombre de sous-items, certaines informations étant absentes. Ils ne pourraient constituer un « Groupe d'association » au sens du langage INFOL d'origine.

On demande de sélectionner les produits pour lesquels des livraisons supérieures à 1 500 unités ont été faites à Marseille dans la première semaine de mai 1967.

Le programme INFOL 2 qui résout ce problème est le suivant :

```

RETRIEVAL CRITERIA 1 *15* ANY EQ MARSEILLE
                      *7* SUB ITEM ASSOCIATED GE
                        1 /MAY 1967 AND LE
                        7 /MAY /1967
                      *13* SUB ITEM ASSOCIATED GT
                        1 500
  
```

EXEMPLE 2

On donne un fichier de répartition géographique des ventes de produits pétroliers pour la France, chaque fiche représentant un mois. Les données

2. sont libérés du service militaire mais n'ont pas été réformés ou exemptés :

RETRIEVAL CRITERIA 7 *9* EQ 4 OR *4* LAST LT 1 500
 8 EQ LIBERE AND NE
 (REFORME OR EXEMPTÉ)

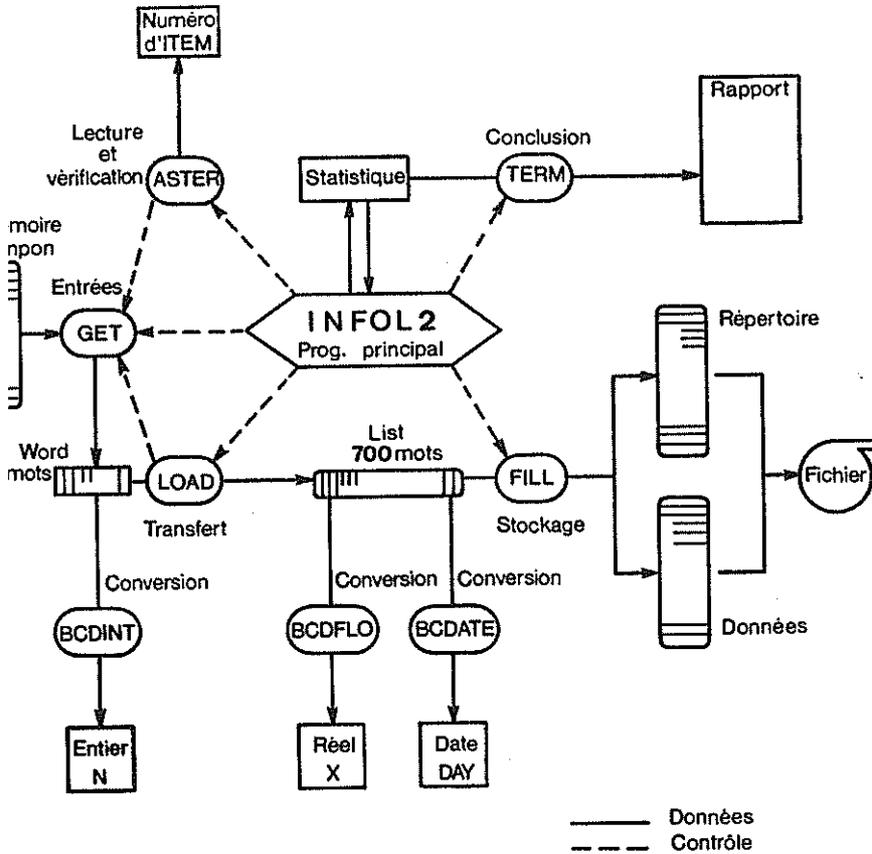


Figure 3

Création d'un fichier avec INFOL 2 : organisation simplifiée du processeur

(Le lecteur pourra vérifier que l'employé n° 689 pris comme exemple figure 1 serait sélectionné par ces critères.) Les deux instructions ci-dessus constituent l'ensemble du « programme » INFOL 2 que l'on doit écrire pour effectuer cette sélection.

de la table ci-dessous montrent des ventes fictives pour un mois donné :

Item 3 Région	QUANTITES VENDUES (m ³)			
	Item 4 Essence	Item 5 Pétrole	Item 6 Super	Item 7 Gas-oil
Nord	23 700	764	26 310	20 641
Picardie	11 850	176	14 170	9 554
Paris	34 500	2 562	80 740	41 496
Centre	18 075	241	22 540	13 570
Limousin	6 725	95	5 520	3 473
Aquitaine	34 500	630	41 850	15 610
Alsace	8 575	165	10 760	9 960
Languedoc	15 040	197	16 100	11 652

On demande un rapport final comprenant uniquement les mois où le maximum de la vente d'essence était supérieure à 20 000 m³ et a été observé à Paris ou dans la région Nord, avec une consommation de supercarburant supérieure à 50 000 m³ dans la même région.

Le programme suivant résoud le problème posé.

```
RETRIEVAL CRITERIA 1 *4* MAXIMUM GT 20 000
                    *3* SUB ITEM ASSOCIATED EQ
                      (PARIS OR NORD)
                    *6* SUB ITEM ASSOCIATED GT
                      50 000
```

On notera qu'il y a deux maxima égaux pour les ventes d'essence dans le mois pris comme exemple (item 4, sous-items 3 et 6). Un seul groupe de valeurs associées vérifie l'ensemble des critères (sous-item 3).

Il existe des différences majeures entre les deux versions d'INFOL au moment de la production du rapport final. La nouvelle version ne comporte pas d'option mais offre une interprétation automatique des informations statistiques accumulées pendant la création du fichier. Le rapport final peut prendre deux formes standard.

1. Si la longueur totale des descriptions ou des données elles-mêmes pour les items à extraire est supérieure à la largeur de la page, le programme organise l'information de manière à l'imprimer verticalement.

2. Dans le cas contraire, les descriptions des items sont imprimées en tête et les valeurs correspondantes sont tabulées horizontalement pour un élément donné.

L'utilisateur conserve cependant le contrôle général puisqu'il peut « forcer » un rapport vertical par l'addition d'items appropriés. Des instructions spéciales existent en INFOL 2 quand on veut obtenir unique-

ment le nombre d'éléments qui obéissent aux critères donnés, ou la liste de leurs indicatifs. Dans la forme ordinaire du rapport, ces informations sont imprimées automatiquement avec les valeurs des items sélectionnés.

ANALYSE CRITIQUE DU LANGAGE. LANGAGES FUTURS

Les langages et systèmes de gestion actuels sont criticables à plusieurs titres : ils ont en commun des défauts qui tiennent à la nature même des problèmes rencontrés et aux limitations technologiques ; d'autre part, certaines déficiences sont liées à l'insuffisance des recherches qui ont été faites dans ce domaine. Pour rester dans le cadre précis des expériences décrites plus haut, c'est-à-dire des tests auxquels le premier processeur INFOL a été soumis, des essais qui ont conduit à la réalisation d'INFOL 2, et des applications faites surtout dans le domaine médical (où des fichiers comprenant près de 200 items par élément ont été implantés) on peut classer en trois catégories les fonctions qui laissent à désirer. Cette liste n'est évidemment pas limitative.

I. — CONCEPTION GENERALE

Dans ses échanges avec l'utilisateur d'une part, avec l'information contenue dans le fichier d'autre part, le langage INFOL pêche sur quatre points importants, que l'on retrouverait dans tous les systèmes de gestion actuels.

1. Technologie

Comme nous l'avons rappelé au début de cet article, le traitement à distance de fichiers utilisés en temps réel se heurte à des problèmes non résolus de conception de systèmes. Les langages de gestion ne deviendront d'utiles instruments de travail que lorsqu'ils pourront fonctionner en mode conversationnel et à distance.

2. Structure des fichiers

Un fichier divisé en éléments, eux-mêmes composés d'items à multiplicité variable, possède une structure suffisamment générale pour satisfaire une majorité d'utilisateurs. Il reste que la structure la plus naturelle pour les applications futures serait une structure d'arbre ou si l'on veut, une structure de liste. Elle permettrait d'établir des fichiers de profondeur théoriquement illimitée.

3. Problème du mode dominant

La responsabilité de déclarer un mode (entier, réel, date, code, alphanumérique, etc.) pour chaque item devrait être retirée à l'utilisateur toutes les fois où cela est possible. La commodité de programmation doit

ment le nombre d'éléments qui obéissent aux critères donnés, ou la liste de leurs indicatifs. Dans la forme ordinaire du rapport, ces informations sont imprimées automatiquement avec les valeurs des items sélectionnés.

ANALYSE CRITIQUE DU LANGAGE. LANGAGES FUTURS

Les langages et systèmes de gestion actuels sont criticables à plusieurs titres : ils ont en commun des défauts qui tiennent à la nature même des problèmes rencontrés et aux limitations technologiques d'autre part, certaines déficiences sont liées à l'insuffisance des recherches qui ont été faites dans ce domaine. Pour rester dans le cadre précis de expériences décrites plus haut, c'est-à-dire des tests auxquels le premier processeur INFOL a été soumis, des essais qui ont conduit à la réalisation d'INFOL 2, et des applications faites surtout dans le domaine médical (où des fichiers comprenant près de 200 items par élément ont été implantés) on peut classer en trois catégories les fonctions qui laissent à désirer. Cette liste n'est évidemment pas limitative.

I. — CONCEPTION GENERALE

Dans ses échanges avec l'utilisateur d'une part, avec l'information contenue dans le fichier d'autre part, le langage INFOL pêche sur quatre points importants, que l'on retrouverait dans tous les systèmes de gestion actuels.

1. Technologie

Comme nous l'avons rappelé au début de cet article, le traitement à distance de fichiers utilisés en temps réel se heurte à des problèmes non résolus de conception de systèmes. Les langages de gestion ne deviendront d'utiles instruments de travail que lorsqu'ils pourront fonctionner en mode conversationnel et à distance.

2. Structure des fichiers

Un fichier divisé en éléments, eux-mêmes composés d'items à multiplicité variable, possède une structure suffisamment générale pour satisfaire une majorité d'utilisateurs. Il reste que la structure la plus naturelle pour les applications futures serait une structure d'arbre ou si l'on veut, une structure de liste. Elle permettrait d'établir des fichiers de profondeur théoriquement illimitée.

3. Problème du mode dominant

La responsabilité de déclarer un mode (entier, réel, date, code, alphanumérique, etc.) pour chaque item devrait être retirée à l'utilisateur toutes les fois où cela est possible. La commodité de programmation doit

la table ci-dessous montrent des ventes fictives pour un mois donné :

Item 3 Région	QUANTITES VENDUES (m ³)			
	Item 4 Essence	Item 5 Pétrole	Item 6 Super	Item 7 Gas-oil
Nord	23 700	764	26 310	20 641
Picardie	11 850	176	14 170	9 554
Paris	34 500	2 562	80 740	41 496
Centre	18 075	241	22 540	13 570
Limousin	6 725	95	5 520	3 473
Aquitaine	34 500	630	41 850	15 610
Alsace	8 575	165	10 760	9 960
Languedoc	15 040	197	16 100	11 652

On demande un rapport final comprenant uniquement les mois où le maximum de la vente d'essence était supérieure à 20 000 m³ et a été observé Paris ou dans la région Nord, avec une consommation de supercarburant supérieure à 50 000 m³ dans la même région.

Le programme suivant résout le problème posé.

RETRIEVAL CRITERIA 1 *4* MAXIMUM GT 20 000
 3 SUB ITEM ASSOCIATED EQ
 (PARIS OR NORD)
 6 SUB ITEM ASSOCIATED GT
 50 000

On notera qu'il y a deux maxima égaux pour les ventes d'essence dans le mois pris comme exemple (item 4, sous-items 3 et 6). Un seul couple de valeurs associées vérifie l'ensemble des critères (sous-item 3).

Il existe des différences majeures entre les deux versions d'INFOL au moment de la production du rapport final. La nouvelle version ne comporte pas d'option mais offre une interprétation automatique des informations statistiques accumulées pendant la création du fichier. Le rapport final peut prendre deux formes standard.

1. Si la longueur totale des descriptions ou des données elles-mêmes pour les items à extraire est supérieure à la largeur de la page, le programme organise l'information de manière à l'imprimer verticalement.

2. Dans le cas contraire, les descriptions des items sont imprimées en tête et les valeurs correspondantes sont tabulées horizontalement pour un élément donné.

L'utilisateur conserve cependant le contrôle général puisqu'il peut forcer un rapport vertical par l'addition d'items appropriés. Des instructions spéciales existent en INFOL 2 quand on veut obtenir unique-

ici céder le pas au traitement interprétatif des données. Supposons par exemple qu'un fichier contienne la phrase :

« LA HAUSSE DES VENTES INDUSTRIELLES S'EST POURSUIVIE D'OCTOBRE 1966 A JUIN 1967 ».

Nous avons ici au sens d'INFOL, conflit entre trois modes : un texte contient des nombres qui doivent être utilisés comme tels puisqu'ils peuvent figurer dans des tests d'inégalité, mais on fait appel à ces nombres sous forme de dates. Comme il n'est pas possible d'anticiper les requêtes adressées à un tel fichier on doit chercher une solution garantissant une généralité complète.

4. Description formelle du langage

Il est rare que les constructeurs ou les groupes d'utilisateurs donnent une description théorique précise des nouveaux langages. Il serait souhaitable que dans le domaine de la gestion, un effort semblable à celui du *Comité Algol* de 1958 soit organisé. Ce comité viserait à donner une base formelle à un langage de grande généralité, ou du moins à en préciser nettement les objectifs. La réalisation d'un tel langage fournirait alors un véritable standard, et non plus de simples résultats expérimentaux.

II. — PROBLEMES D'INTERFACE

Les performances d'un langage qui travaille (comme INFOL) dans un environnement verbal doivent être évaluées en termes de richesse des échanges et non plus seulement en termes de calcul et de précision. La plupart des utilisateurs de systèmes se contentent actuellement de l'extraction brute d'éléments de fichiers obéissant à des critères simples. INFOL permet l'extraction d'un élément sur la base de sa structure (présence ou absence d'items, valeurs dérivées comme la valeur moyenne des sous-items, etc.). Dans les langages futurs, on voudra sans doute répondre à des questions qui feront appel à trois fonctions nouvelles :

i) la possibilité d'augmenter le fichier en stockant temporairement des valeurs dérivées,

ii) le fait de calculer ces nouvelles valeurs par des opérations tirées d'une bibliothèque standard (qui pourront être déclarées par l'utilisateur soit au moment de la création du fichier, soit au moment de l'exécution),

iii) l'exécution de *verbes*, eux aussi représentés sous forme de sous-programmes internes ou externes.

III. — PROGRAMMATION

1. Protection des items à caractère confidentiel

Le langage doit prévoir la protection globale du fichier et doit aussi assurer que certains items ne peuvent être obtenus en clair que par un utilisateur autorisé (secret professionnel dans les fichiers légaux ou médicaux, applications militaires, etc.).

2. Révision dynamique des fichiers

La tâche de réviser la structure d'un fichier donné pourrait devenir une fonction intégrée du langage ; au fur et à mesure que des items seraient créés ou des mises à jour réalisées, le fichier évoluerait au moyen d'un chaînage interne.

Un problème d'optimisation de la stratégie de la recherche se pose donc.

CONCLUSION

Un vaste domaine d'application s'ouvre pour les langages de gestion qui permettront de faire l'économie des nombreux programmes spécialisés actuellement indispensables au traitement des fichiers et de plus de les interroger en temps réel. Certes la documentation automatique sur la base du langage naturel n'est possible actuellement que dans des expériences limitées (3). D'un autre côté, l'évolution de la technologie des ordinateurs va imposer un mode de traitement qui dépasse les méthodes actuelles. Dans certains domaines d'importance cruciale, il est impératif d'avoir un accès continu à des fichiers tenus à jour et ce dans un langage de haut niveau. Cette contrainte impose l'absence de programmation intermédiaire et l'élimination des servitudes élémentaires (lecture de bandes, spécification de formats pour les rapports, etc.). En d'autres termes l'ordinateur lui-même doit être doté de langages plus puissants que ceux dont nous disposons actuellement. Ces futurs langages de gestion chercheront un « juste milieu » entre le vocabulaire naturel et les systèmes à cartes paramètres dont la vulnérabilité est trop grande.

Nous avons passé en revue quelques expériences prenant pour base un langage simple obéissant à ces critères. On peut se demander maintenant quels objectifs lointains peuvent être assignés aux ordinateurs dans ce domaine. Des discussions ont eu lieu récemment aux États-Unis (4) pour définir un « langage » économique et financier qui répondrait à des questions en anglais utilisant un vocabulaire technique, par exemple :

— Calculer l'effet d'une chute de 5% en produit national brut sur les gains par action pour toutes les valeurs automobiles.

— Classer les compagnies cotées dans l'ordre croissant des gains sur leurs actions et calculer la variance de cette distribution autour de la croissance moyenne.

— Calculer le coefficient de régression entre les ventes et les valeurs du produit national brut pour U.S. Steel dans la période 1952-1963.

L'expérience d'INFOL montre à notre avis que la conception d'un processeur de ce type devra se faire sur le principe d'une *hiérarchie de langages formels*. Les entités linguistiques d'origine déclencheront une série de requêtes codées semblables aux « critères de sélection » d'INFOL, eux-mêmes susceptibles d'être traduits dans la notation d'une application donnée. La difficulté majeure se trouve alors dans la définition d'un langage d'entrée sans équivoque, avec toutefois une sensibilité suffisante

2. Révision dynamique des fichiers

La tâche de réviser la structure d'un fichier donné pourrait devenir une fonction intégrée du langage ; au fur et à mesure que des iter seraient créés ou des mises à jour réalisées, le fichier évoluerait au moyen d'un chaînage interne.

Un problème d'optimisation de la stratégie de la recherche se pose donc.

CONCLUSION

Un vaste domaine d'application s'ouvre pour les langages de gestion qui permettront de faire l'économie des nombreux programmes spécialisés actuellement indispensables au traitement des fichiers et de plus à les interroger en temps réel. Certes la documentation automatique sur la base du langage naturel n'est possible actuellement que dans des expériences limitées (3). D'un autre côté, l'évolution de la technologie des ordinateurs va imposer un mode de traitement qui dépasse les méthodes actuelles. Dans certains domaines d'importance cruciale, il est impératif d'avoir un accès continu à des fichiers tenus à jour et ce dans un langage de haut niveau. Cette contrainte impose l'absence de programmation intermédiaire et l'élimination des servitudes élémentaires (lecture de bandes, spécification de formats pour les rapports, etc.). En d'autres termes l'ordinateur lui-même doit être doté de langages plus puissants que ceux dont nous disposons actuellement. Ces futurs langages de gestion chercheront un « juste milieu » entre le vocabulaire naturel et les systèmes à cartes paramètres dont la vulnérabilité est trop grande.

Nous avons passé en revue quelques expériences prenant pour base un langage simple obéissant à ces critères. On peut se demander maintenant quels objectifs lointains peuvent être assignés aux ordinateurs dans ce domaine. Des discussions ont eu lieu récemment aux États-Unis (4) pour définir un « langage » économique et financier qui répondrait à de nombreuses questions en anglais utilisant un vocabulaire technique, par exemple

— Calculer l'effet d'une chute de 5% en produit national brut sur les gains par action pour toutes les valeurs automobiles.

— Classer les compagnies cotées dans l'ordre croissant des gains sur leurs actions et calculer la variance de cette distribution autour de la croissance moyenne.

— Calculer le coefficient de régression entre les ventes et les valeurs du produit national brut pour U.S. Steel dans la période 1952-1963.

L'expérience d'INFOL montre à notre avis que la conception d'un processeur de ce type devra se faire sur le principe d'une *hiérarchie de langages formels*. Les entités linguistiques d'origine déclencheront une série de requêtes codées semblables aux « critères de sélection » d'INFOL, eux-mêmes susceptibles d'être traduits dans la notation d'une application donnée. La difficulté majeure se trouve alors dans la définition d'un langage d'entrée sans équivoque, avec toutefois une sensibilité suffisante

céder le pas au traitement interprétatif des données. Supposons par exemple qu'un fichier contienne la phrase :

« LA HAUSSE DES VENTES INDUSTRIELLES S'EST POURSUIVIE D'OCTOBRE 1966 A JUIN 1967 ».

Nous avons ici au sens d'INFOL, conflit entre trois modes : un texte contient des nombres qui doivent être utilisés comme tels puisqu'ils peuvent figurer dans des tests d'inégalité, mais on fait appel à ces nombres sous forme de dates. Comme il n'est pas possible d'anticiper les requêtes adressées à un tel fichier on doit chercher une solution garantissant une généralité complète.

Description formelle du langage

Il est rare que les constructeurs ou les groupes d'utilisateurs donnent une description théorique précise des nouveaux langages. Il serait souhaitable que dans le domaine de la gestion, un effort semblable à celui du Comité Algol de 1958 soit organisé. Ce comité viserait à donner une description formelle à un langage de grande généralité, ou du moins à en préciser nettement les objectifs. La réalisation d'un tel langage fournirait alors un véritable standard, et non plus de simples résultats expérimentaux.

II. — PROBLEMES D'INTERFACE

Les performances d'un langage qui travaille (comme INFOL) dans un environnement verbal doivent être évaluées en termes de richesse des langages et non plus seulement en termes de calcul et de précision. La plupart des utilisateurs de systèmes se contentent actuellement de l'extraction brute d'éléments de fichiers obéissant à des critères simples. INFOL permet l'extraction d'un élément sur la base de sa structure (présence ou absence d'items, valeurs dérivées comme la valeur moyenne des sous-items, etc.). Dans les langages futurs, on voudra sans doute répondre à des questions qui feront appel à trois fonctions nouvelles :

i) la possibilité d'augmenter le fichier en stockant temporairement des valeurs dérivées,

ii) le fait de calculer ces nouvelles valeurs par des opérations tirées d'une bibliothèque standard (qui pourront être déclarées par l'utilisateur soit au moment de la création du fichier, soit au moment de l'exécution),

iii) l'exécution de *verbes*, eux aussi représentés sous forme de sous-programmes internes ou externes.

III. — PROGRAMMATION

Protection des items à caractère confidentiel

Le langage doit prévoir la protection globale du fichier et doit aussi assurer que certains items ne peuvent être obtenus en clair que par un utilisateur autorisé (secret professionnel dans les fichiers légaux ou médicaux, applications militaires, etc.).

au contexte. La détection des ambiguïtés dans l'information déclarative fournie au programme processeur aura donc, à ce niveau, une importance qui n'existait pas dans nos expériences avec INFOL.

1. INFOL. Reference Manual, CDC Publ. No-60170300.
2. VALLEE J, F., CHALICE R. et MITTMAN B., « A Progress Report on INFOL 2 » Northwestern University Report, nov. 1967.
3. VALLEE J. F., KRULEE G. K. et GRAU A. A., « Retrieval Formulae for Inquiry systems » *Information storage and Retrieval Journal* March 1968.
4. Une réunion patronnée par la compagnie IBM s'est tenue sur ce thème à l'Université Northwestern (Evanston, Illinois) en juillet 1966. Elle réunissait des financiers et des théoriciens du software.
5. LAMSON B. G. et DIMSDALE B., « A Natural Language Information Retrieval System » *Proceedings of the IEEE* vol. 54, n° 12, déc. 1966.

11

11