



NUMERO DE PUBLICATION : 1003613A6

NUMERO DE DEPOT : 9101153

Classif. Internat.: G06F

MINISTERE DES AFFAIRES ECONOMIQUES

Date de délivrance : 05 Mai 1992

---

Le Ministre des Affaires Economiques,

Vu la loi du 28 Mars 1984 sur les brevets d' invention, notamment l' article 22;

Vu l' arrêté royal du 2 Décembre 1986 relatif à la demande, à la délivrance et au maintien en vigueur des brevets d' invention, notamment l' article 28;

Vu le procès verbal dressé le 19 Décembre 1991 à 11h20  
à l' Office de la Propriété Industrielle

## ARRETE :

ARTICLE 1.- Il est délivré à : KENT (PATENTS) LIMITED  
Upper Mallow Street 18, LIMERICK(IRLANDE)

représenté(e)(s) par : GOEGEBEUR Erik, BUGNION S.A., Rue de Namur, 43 bte 3 - B  
1000 BRUXELLES.

un brevet d' invention d' une durée de 6 ans, sous réserve du paiement des taxes annuelles, pour : PROCEDE POUR PRODUIRE ET FAIRE FONCTIONNER DES SYSTEMES DE BASES DE DONNEES.

INVENTEUR(S) : Eustace Joseph, c/o Enterprise Centre, North Mall, Cork (IE)

ARTICLE 2.- Ce brevet est délivré sans examen préalable de la brevetabilité de l' invention, sans garantie du mérite de l' invention ou de l' exactitude de la description de celle-ci et aux risques et périls du(des) demandeur(s).

Bruxelles, le 05 Mai 1992  
PAR DELEGATION SPECIALE :

  
WUYTS L.  
Directeur.

PROCEDES POUR PRODUIRE ET FAIRE FONCTIONNER DES SYSTEMES  
DE BASES DE DONNEES

1           La présente invention concerne des procédés  
pour produire et faire fonctionner des systèmes de bases  
de données qui comprennent des structures de mise en  
mémoire pour les données et des processeurs pour mettre  
5 en mémoire, reprendre et traiter les données et établir  
une interface avec un utilisateur.

Certains des impératifs les plus importants  
pour un système de bases de données comprennent les  
éléments suivants :

10

1.           Il ne peut pas y avoir d'erreurs de programme  
provoquant une interruption du fonctionnement  
du système ou la dégradation des données.
  
- 15   2.           Si des défauts se produisent, par exemple par  
une utilisation incorrecte du système, ces  
défauts doivent être facilement détectés et  
éliminés.
  
- 20   3.           Il doit être possible de mettre le système  
facilement à jour, afin d'y inclure des  
matériels et des logiciels récemment mis au  
point.

1

4. Le texte affiché pendant le fonctionnement doit être clair pour un utilisateur, et la production de systèmes de bases de données à utiliser dans des pays ayant des langues différentes doit pouvoir être effectuée de manière précise et efficace.

5

10

Les procédés pour produire un système de bases de données doivent satisfaire à ces impératifs, tout en permettant d'utiliser efficacement le temps de mise au point par les techniciens.

15

20

25

30

35

Jusqu'à présent, les systèmes de bases de données ont été mis au point en subdivisant une structure de dispositif de mise en mémoire et en attribuant des parties de la structure à la mise en mémoire de différentes bases de données. Les programmes sont rédigés ensuite afin de mettre en mémoire, de reprendre et de traiter les données en accédant au dispositif de mise en mémoire. Si un nouveau système de bases de données est mis au point ultérieurement et constitue une amélioration du système initial, la mise à jour du système initial implique le déplacement de toutes les données vers un dispositif de mise en mémoire séparé et la reconfiguration des données en vue de la mise en mémoire, conformément à la nouvelle structure du dispositif de mise en mémoire. Ceci peut provoquer la dégradation des données. Pour ce qui concerne les programmes, ceux-ci sont généralement simplement remplacés par le nouvel ensemble de programmes. Ceci est une opération qui prend du temps et donne souvent lieu à un système qui ne semble pas familier pour l'utilisateur et nécessite donc un certain degré de remise au courant. Compte tenu de ces difficultés pour mettre les systèmes de bases de données à jour, ce travail n'est souvent pas effectué. Par

1 conséquent, un utilisateur ne profite pas souvent du  
bénéfice complet de ces améliorations.

Si un défaut se produit par suite d'un circuit  
mal fabriqué ou d'une erreur de programme, un système  
5 de détection d'erreur tel celui décrit dans le brevet  
européen n° 180299B (ICL) peut être utilisé. Toutefois,  
lorsque les techniciens recherchent le défaut, il est  
très difficile d'identifier exactement où celui-ci se  
produit, afin de permettre sa réparation. Par exemple,  
10 si un appareil d'affichage est vérifié en utilisant le  
procédé décrit dans le brevet européen n° 52,684B (IBM),  
il faudra beaucoup de temps pour identifier le défaut.

Les systèmes de bases de données sont  
généralement produits en vue de l'affichage et de  
15 l'impression d'un texte dans la langue d'un pays déterminé  
et, pour pouvoir utiliser efficacement le système de  
bases de données dans un pays différent, le texte doit  
être traduit. Si le texte affiché est traduit par la  
machine, la signification exacte du texte original n'est  
20 souvent pas exactement rendue. Si le texte est traduit  
manuellement, il peut se produire des discordances et  
cette opération de traduction manuelle demande beaucoup  
de temps. Quoique le texte affiché n'affecte pas  
directement le fonctionnement d'un système de bases de  
25 données, il est extrêmement important parce qu'il permet  
aux utilisateurs (dont beaucoup ne sont pas compétents  
en électronique) d'ordonner l'exécution de la gamme des  
fonctions disponibles en utilisant le système de bases  
de données. Si le texte affiché n'est pas clair, le  
30 système de bases de données sera fortement sous-utilisé.

L'invention vise à proposer des procédés pour  
produire et faire fonctionner des systèmes de bases de  
données à mettre à jour, en surmontant au moins certaines  
de ces difficultés.

35 Conformément à l'invention, il est prévu un

1 procédé pour produire et faire fonctionner un système  
de bases de données, qui comprend les étapes suivantes:

5 détermination des impératifs propres au système  
de bases de données;

conception d'une structure de mise en mémoire de  
la base de données et utilisation de cette conception  
pour produire une structure de mise en mémoire de  
10 bases de données de référence en effectuant les  
opérations suivantes :

15 produire et mettre en mémoire une structure  
de cellules de mémoire vierges pour chacune  
des différentes zones de données et mise en  
mémoire d'un indicateur indiquant quels sont  
les types de caractères autorisés pour la mise  
en mémoire des données dans les cellules;

20 produire et mettre en mémoire un fichier  
d'adresses pour les structures de zones;

25 afficher une représentation d'une structure  
de mise en mémoire de bases de données qui  
comprend plusieurs structures de zones de  
données et vérifier celle-ci après modification  
si nécessaire; et

30 mettre en mémoire la structure de base de  
données vérifiée comme structure de base de  
données de référence dans un dispositif de  
mémoire permanente;

35 duplication de la structure de la base de données  
de référence afin d'obtenir une structure de mise

- 1 en mémoire de base de données active;
- production d'une liste de fonctions de bases de données pour mettre en mémoire, reprendre et traiter
- 5 les données dans la structure de mise en mémoire de la base de données de référence;
- enregistrement des conditions autorisées régissant les opérations de mise en mémoire, de reprise et
- 10 de traitement;
- enregistrement des évènements requis pour mettre le système à l'arrêt en cas de défaut grave, tel que déterminé par l'ensemble des conditions
- 15 autorisées;
- production d'un code pour définir une structure de point de rupture pour traiter les contenus de l'état et de la mémoire;
- 20 production d'un code pour mettre en mémoire un fichier des données de points de rupture;
- conformément à la liste des fonctions, production de plusieurs modules de code de source général sans texte;
- 25 insertion d'un texte dans la langue du pays de destination dans le code de source général, afin de produire un code de source final;
- 30 compilation du code de source final;
- intégration des modules;
- 35

1 vérification du code de machine; et

après utilisation du système de bases de  
données, modification de celui-ci pour améliorer  
5 ses performances en utilisant le matériel et  
le logiciel disponibles.

L'étape de modification du système comprend de  
préférence les sous-étapes suivantes :

10 modification de la structure de mise en mémoire  
de la base de données de référence;

comparaison de la structure de référence  
15 modifiée avec la structure active;

production de données de différence; et

20 modification automatique de la structure active  
afin de la mettre en conformité avec la  
structure de référence modifiée.

L'étape de modifier le système comprend de préférence  
les sous-étapes suivantes :

25 modifier la structure de mise en mémoire de  
la base de données de référence;

30 comparer la structure de référence modifiée  
à la structure active;

produire des données de différence; et

35 modifier automatiquement la structure active  
pour la mettre en conformité avec la structure

1 de référence modifiée.

Dans un mode de réalisation, l'étape de  
production des modules de code de source général comprend  
5 les sous-étapes suivantes :

- insertion d'un indicateur dans le code de source  
indiquant la présence d'un texte;
- 10 insertion d'une marque indiquant que le texte ne  
doit pas être traduit:  
  
affichage séquentiel de chaque élément de texte  
à traduire;
- 15 optimisation de la disposition du texte;  
  
vérification de la disposition du texte;
- 20 exécution des opérations suivantes sur chacune des  
lignes du code de source :  
  
vérification de la présence d'une marque de  
texte;
- 25 conformément à cette constatation, extraction  
du texte à traduire, s'il y a lieu;  
  
comparaison du texte avec tous les textes  
préalablement extraits;
- 30 assignation d'un nouveau code de texte si celui-  
ci n'a pas déjà été extrait;  
  
reprise d'un code de texte associé s'il a déjà
- 35

1           été extrait; et

          insertion du code de texte repris et assigné  
          dans la ligne de code de source.

5

          Dans un autre mode de réalisation, l'étape  
d'insertion du texte dans le module de code de source  
général comprend les sous-étapes suivantes :

10           production d'une base de données de traduction ayant  
          un enregistrement de texte traduit pour chaque code  
          de texte, une base de données de traduction existant  
          pour chaque langue;

15           production d'un ensemble de codes de machines pour  
          un programme dans une langue, en exécutant les étapes  
          suivantes :

          filtrage de chaque ligne de code de source  
20           et extraction d'un code de texte, s'il existe;

          utilisation du code de texte pour reprendre  
          le texte traduit correspondant pour la base  
          de données de traduction appropriée;

25

          insertion du texte traduit dans la ligne de  
          code de source à la place du code de texte;  
          et

30           compilation du code de source.

          L'invention sera mieux comprise à l'examen  
de la description ci-après de certains de ses modes de  
réalisation préférés, donnés uniquement à titre d'exemple  
35           et en se référant aux figures en annexe, qui représentent

1 respectivement :

La figure 1, un diagramme synoptique montrant un procédé de production et de fonctionnement d'un système de bases de données;

5 La figure 2, un diagramme synoptique montrant une partie du procédé avec plus de détails;

La figure 3, un diagramme synoptique montrant des procédés pour faire fonctionner, et notamment pour mettre à jour le système produit;

10 Les figures 4(a) et 4(b), des diagrammes synoptiques plus détaillés pour une partie du procédé de production du système; et

Les figures 5(a), 5(b) et 5(c), des illustrations d'exemples de lignes de code de source préliminaires, générales et finales produites par le procédé de la présente invention.

Si on examine la figure 1, on peut y voir un procédé pour produire un système de bases de données. Le procédé est généralement désigné par le numéro de référence 1 et comprend une étape initiale 2 consistant à déterminer les impératifs pour le système de bases de données. Plus particulièrement, cette étape comprend la détermination des données à mettre en mémoire dans le système et les index pour lesquels des recherches doivent être effectuées afin de reprendre et de trier les données. Ceci conduit à l'étape 3 pour la conception d'une structure de mise en mémoire de bases de données qui est conforme à tous les impératifs. La conception de la structure de mise en mémoire comprend des spécifications pour le disque fixe, le circuit de mémoire, la largeur de bus et le microprocesseur.

Après l'étape 3, une structure de mise en mémoire de bases de données de référence est produite à l'étape 4, cette structure comprenant des valeurs de

1 paramètres définissant les structures des zones de données  
pour la mise en mémoire dans un dispositif de mise en  
mémoire. L'étape 4 sera décrite plus en détail ci-après.  
A l'étape 5, un code de détection d'erreur est produit  
5 et est incorporé à des signaux qui concernent la structure  
de mise en mémoire de la base de données de référence.  
L'étape 5 sera également décrite plus en détail ci-après.  
Parallèlement avec l'étape 5, on effectue l'étape 8 qui  
comprend la production de modules de code de source  
10 général. Un module est un groupe de lignes de code de  
source correspondant qui ne sont pas propres au texte  
et à la langue d'un pays particulier. D'une manière  
générale, chaque module concerne un traitement particulier  
tel que la reprise de données, la mise en mémoire des  
15 données, le triage, l'affichage ou certaines autres  
opérations. Après les étapes 5 et 8, le texte pour la  
langue d'un pays déterminé est inséré à l'étape 9 dans  
les modules de code de source général. A l'étape 10,  
le code de source de chaque module est compilé afin de  
20 produire un code de machine. A l'étape 11, les modules  
sont intégrés les uns aux autres par mise en mémoire  
dans des emplacements d'un dispositif de mise en mémoire,  
si bien qu'ils peuvent subir une vérification croisée  
et être incorporés au matériel et au logiciel. A l'étape  
25 12, le code de machine compilé est vérifié et le système  
est alors prêt à être utilisé. A une certaine date  
ultérieure à l'étape 13, le système est modifié afin  
d'y incorporer le matériel et le logiciel améliorés.

A la figure 1, les sous-étapes 20 à 24  
30 représentent l'étape 5 comprenant la production d'un  
code de détection d'erreurs défini plus en détail. A  
l'étape 20, on enregistre un ensemble de conditions  
autorisées pour le fonctionnement du système de bases  
de données. Ces conditions autorisées comprennent les  
35 dimensions des registres de mémoire, une gamme de vitesses

1 de traitement, des capacités de mémoire souhaitées et  
d'autres conditions. Ces conditions sont produites  
conformément aux paramètres de la structure de référence  
produite à l'étape 4. A l'étape 21, on enregistre des  
5 évènements d'arrêt prédéterminés, ces évènements étant  
ceux en réponse auxquels le système de bases de données  
devrait cesser de fonctionner. Un exemple d'évènement  
de ce genre est une erreur de parité qui indique la  
dégradation des données ou un défaut répété empêchant  
10 l'accès à un dispositif de mise en mémoire particulier.  
A l'étape 22, le code de distribution du point de rupture  
est produit afin d'ordonner à un processeur d'enregistrer  
le contenu du circuit de mémoire et l'emplacement de  
l'indicateur de programme si une erreur se produit. Le  
15 code produit à l'étape 22 entre uniquement en  
fonctionnement s'il se produit une erreur peu importante  
qui, par exemple, ne peut pas produire une dégradation  
des données. A l'étape 23, un manipulateur de fichier  
de point de rupture est produit afin de permettre  
20 la mise en mémoire des données de distribution de point  
de rupture dans un fichier unique mis en mémoire dans  
une mémoire permanente. L'étape 24 comprend la production  
d'instructions de mise à l'arrêt pour un processeur,  
afin d'interrompre le fonctionnement du système si on  
25 détecte un évènement enregistré à l'étape 21. Le code  
de détection d'erreur produit aux étapes 20 à 24 est  
incorporé à un ou plusieurs des modules de code de source  
général.

L'effet du code produit à l'étape 5 consiste en  
30 ce que, s'il se produit une erreur grave donnant lieu à  
dégradation des données, le système de bases de données  
est automatiquement mis à l'arrêt d'une manière contrôlée  
(en utilisant le code des étapes 21 et 24), ce qui ne  
provoque pas la perte d'autres données que celles qui ont  
35 déjà été perdues. Le code produit à l'étape 22 ordonne

1 à un processeur de mettre en mémoire tout le contenu  
de la mémoire et des registres d'un microprocesseur dans  
un dispositif de mise en mémoire, si bien que celui-ci  
peut être recherché à une date ultérieure. Ces  
5 instructions sont données avant la mise à l'arrêt du  
système. Toutefois, si une erreur sans gravité se produit,  
telle que la mise en mémoire de données dans un dispositif  
de mise en mémoire non autorisé, le processeur du système  
de bases de données recevra ordre de produire des données  
10 de point de rupture qui donnent une image du contenu  
de la mémoire et du fonctionnement du programme au moment  
où l'erreur s'est produite. Ces données de point de  
rupture sont mises en mémoire dans un fichier unique  
(en utilisant le code de l'étape 23) dans une mémoire  
15 permanente, de façon à pouvoir être facilement retrouvées  
à une date ultérieure en vue de leur examen par les  
techniciens. De cette manière, la source d'une erreur  
peut être immédiatement identifiée sans effectuer des  
opérations d'analyse compliquées. Ceci constitue un aspect  
20 extrêmement important de la présente invention vu que  
ceci permet l'identification immédiate d'erreurs  
relativement peu importantes, si bien que celles-ci  
peuvent être éliminées avec une courte interruption du  
fonctionnement du système de bases de données et tout  
25 en assurant en même temps la mise à l'arrêt du système  
d'une manière contrôlée si une erreur grave se produit.

Si on examine maintenant la figure 2, on peut  
y voir plus en détail l'étape 4 pour la production de  
la structure de mise en mémoire de bases de données de  
30 référence. A l'étape 3, on met en mémoire une structure  
de mémoire des cellules de mémoire vierges pour chaque  
zone de données. La structure de mémoire comprend des  
bytes de mémoire pour chaque zone de données, l'adresse  
souhaitée des bytes et l'emplacement des bytes par rapport  
35 à d'autres zones de la structure de la base de données.

1 A l'étape 31, on met en mémoire dans le dispositif de  
mise en mémoire une indication des caractères de données  
autorisés pouvant être introduits dans la structure de  
la mémoire de zone. Dans la plupart des cas, cette  
5 indication est alphanumérique. A l'étape 32, on produit  
un fichier d'adresses qui est également mis en mémoire  
dans le dispositif de mise en mémoire et qui indique  
les adresses de tous les groupes des zones correspondantes  
dans une base de données. A l'étape 33, la structure de  
10 bases de données est vérifiée par vérification manuelle  
quant à la cohérence et aux références correctes et quant  
à l'aptitude à la mise en mémoire du type particulier  
de données qui est requis. A l'étape 4, toutes les  
structures de mémoire sont mises en mémoire ensemble  
15 sous forme d'une structure de bases de données de  
référence, désignée par le numéro 35. La structure de  
bases de données est ensuite reproduite de façon à ce  
qu'une structure soit mise en mémoire pour les références  
futures, tandis que l'autre est utilisée comme structure  
20 active pour le système en cours de préparation. A l'étape  
36, qui est effectuée en parallèle avec l'étape 34, on  
produit une liste fonctionnelle indiquant les fonctions  
des programmes requis pour la mise en mémoire, la reprise  
et le traitement des données mises en mémoire dans la  
25 structure de la base de données de référence. C'est  
d'après la liste fonctionnelle que les modules de code  
de source général sont produits à l'étape 8.

La structure de mise en mémoire de la base  
de données actives qui a été produite est celle qui est  
30 utilisée pour le système de bases de données en cours  
de préparation. En enregistrant successivement chaque  
structure de mémoire de zone et en interconnectant  
ultérieurement les structures de zones et en vérifiant  
la structure de bases de données résultante, on obtient  
35 que la structure de la base de données s'adapte

1 généralement de manière idéale aux impératifs de  
l'utilisateur et permet une utilisation très efficace  
de la mémoire. Cette solution s'est avérée  
particulièrement appropriée pour utiliser efficacement  
5 tout l'espace de mémoire, parce que des réserves peuvent  
être prévues pour enregistrer des structures de mémoire  
de zones supplémentaires sans nécessiter la mise en  
mémoire d'une structure de bases de données dans plus  
d'un dispositif de mise en mémoire.

10 La figure 3 représente plus en détail l'étape  
13 de la modification du système mis au point. La  
modification sera effectuée après une utilisation  
considérable du système. Comme mentionné ci-dessus, si  
la structure de bases de données de référence produite  
15 est redoublée, on prévoit une structure à utiliser comme  
structure de bases de données actives du système et  
l'autre structure comme structure de bases de données  
de référence.

A l'étape 40, la structure de bases de données  
20 de référence est modifiée afin de s'adapter aux nouveaux  
impératifs de l'utilisateur et au matériel et aux  
programmes nouvellement disponibles. A l'étape 41, si  
une instruction est reçue pour le système afin de mettre  
à jour la base de données actives et de référence, les  
25 structures mises en mémoire sont automatiquement comparées  
par un processeur. Ceci implique la comparaison des  
paramètres utilisés pour établir les structures de bases  
de mémoire, y compris l'espace de mémoire et les types  
de caractères autorisés. A l'étape 43, on produit un  
30 fichier indiquant les différences entre les structures  
de mise en mémoire des bases de données actives et de  
référence. Parallèlement à l'étape 43, les données de  
la structure de mise en mémoire de la base de données  
actives sont transférées à l'étape 42 à une unité de  
35 disque séparée. Le processeur met temporairement en

1 mémoire le fichier des différences et modifie  
automatiquement la structure de mise en mémoire de la  
base de données actives, si bien que celle-ci est  
semblable à la structure de référence. A l'étape 45,  
5 les listes fonctionnelles sont comparées et, à l'étape  
46, les données qui ont été transférées préalablement  
sont alors stockées dans la structure de mise en mémoire  
de la base de données actives modifiée. Toutes  
modifications requises aux programmes sont également  
10 effectuées en fonction de la comparaison entre les listes  
fonctionnelles.

On comprendra que, si on modifie de cette manière  
le système de bases de données, on peut maintenir une  
seule structure de mise en mémoire de bases de données  
15 de référence à un emplacement central, tandis que des  
structures de mises en mémoire actives identiques sont  
utilisées par différents utilisateurs en différents  
emplacements. Lorsque le système est modifié, cette  
opération est réalisée à l'emplacement central en  
20 modifiant la structure de référence. La structure de  
mise en mémoire nouvelle de la base de données de  
référence est ensuite comparée à toutes les structures  
actives et les modifications sont effectuées  
automatiquement. De cette manière, on peut obtenir que  
25 toutes les structures de bases de données actives soient  
automatiquement mises à jour et mises en conformité avec  
la dernière version. On a constaté que la structure de  
mise en mémoire de bases de données constitue un élément  
clé essentiel pour la modification du système de bases  
30 de données et, grâce à une comparaison automatique et  
à la régénération d'une nouvelle structure de mise en  
mémoire active, les défauts sont éliminés. Il est  
évidemment bien important également de veiller à ce que  
les programmes soient modifiés automatiquement,  
35 parallèlement avec cette opération. En pratique, si une

1 erreur se produit dans un système de bases de données  
actives, le système est automatiquement mis à jour si  
ceci n'a pas déjà été fait, si bien que l'appareillage  
de vérification peut être utilisé pour ce système  
5 particulier. Ceci évite la nécessité de disposer d'un  
grand nombre de systèmes de vérification et de  
reconfigurer ceux-ci.

Si on examine maintenant simultanément les  
figures 4(a) et 4(b), on peut y voir plus en détail les  
10 étapes 8 et 9 du processus représenté à la figure 3.  
Ces étapes impliquent la production de modules de code  
de source général et l'insertion du texte d'un pays  
déterminé dans les modules. A l'étape 50, on enregistre  
des instructions pour une ligne de code de source . Ces  
15 instructions peuvent concerner la reprise des données  
ou certaines autres opérations effectuées, comme indiqué  
par la liste fonctionnelle produite à l'étape 36. A  
l'étape 51, on décide si un texte est requis ou non pour  
la ligne de code de source et, si ce n'est pas le cas,  
20 des instructions sont mises en mémoire à l'étape 57 sous  
forme de ligne de code de source préliminaire et le  
processus est répété pour la ligne de code de source  
suivante. Toutefois, si un texte est requis, le texte  
est inséré à l'étape 53 dans la ligne de code de source  
25 avec un indicateur indiquant que le texte est présent  
dans la ligne. A l'étape 54, on décide si le texte doit  
être utilisé ou non pour un panneau d'affichage dans  
un système de bases de données, c'est-à-dire s'il doit  
être examiné par un utilisateur. Dans la majorité des  
30 cas, le texte sera examiné par un utilisateur et , dans  
ces cas, l'étape 5 consistant à optimiser la ligne de  
code de source et à la vérifier est mise en oeuvre.  
L'optimisation de la ligne implique l'affichage du texte  
dans le contexte des autres données qui seront affichées  
35 sur l'écran. Ceci permet à un concepteur de s'assurer

1 que le texte sera clairement affiché et sera en accord  
avec le reste de l'affichage sur écran, quelle que soit  
la langue du texte. Toutefois, si le texte ne doit pas  
être affiché, une marque est insérée à l'étape 56 dans  
5 la ligne de code de source pour indiquer cette situation.  
Les étapes 50 à 57 sont répétées pour chaque ligne de  
code de source requise pour chaque module.

La figure 5(a) représente un exemple de  
certaines lignes de code de source préliminaires mises  
10 au point maintenant. Le texte est en langue anglaise  
parce que le système est produit dans un pays de langue  
anglaise. Les indicateurs de texte sont `TEXT` et  
`ENDTEXT` et la marque indiquant qu'une traduction n'est  
pas requise est `(SYSTEXT)`. La marque est dans la  
15 section de "commentaires" de la ligne, tandis que les  
indicateurs de texte sont des instructions de  
programmation dans la partie principale de la ligne.

A ce stade, on met en mémoire un ensemble de  
lignes de code de source préliminaires et, à l'étape  
20 60, la première de ces lignes est introduite dans la  
mémoire, est vérifiée automatiquement pour détecter la  
présence du texte et, à l'étape 61, on décide si le texte,  
au cas où il existe, doit être traduit ou non. Ceci  
implique une vérification de l'existence ou de l'absence  
25 d'un indicateur de texte. Si celui-ci existe, on vérifie  
s'il existe une marque. S'il n'y a pas de marque, c'est  
alors que le texte doit être traduit. Toutefois, s'il  
existe une marque, le texte n'est pas à traduire et est  
prêt à être utilisé par les techniciens de développement.

30 A l'étape 62, la ligne de code de source est mise en  
mémoire sous forme de ligne de source générale et le  
processus est répété ultérieurement jusqu'à l'étape de  
décision 63. Si le texte est destiné à la traduction,  
un processeur extrait automatiquement le texte à l'étape  
35 64, en détectant les indicateurs de texte et en

1 enregistrant sur un fichier séparé tous les bytes contenus  
dans les indicateurs. A l'étape 65, le processeur produit  
un code pour cette partie de texte. L'étape 65 est assez  
complexe car elle implique la comparaison de la ligne  
5 de texte en cours avec toutes les lignes de texte  
extraites précédemment. Si elle n'a pas été extraite  
précédemment, un nouveau code lui est assigné. Si elle  
a été extraite précédemment, le code qui avait été utilisé  
antérieurement est utilisé à nouveau. A l'étape 66, le  
10 code de texte nouvellement produit ou repris est inséré  
dans la ligne de code de source afin de produire une  
ligne de code de source générale qui est mise en mémoire  
à l'étape 62. Cette opération est répétée à nouveau pour  
chacune des lignes de code de source préliminaires. A  
15 la fin de ces opérations, il existe un ensemble de lignes  
de code de source générales qui ne comprennent aucun  
texte destiné à l'affichage mais simplement une série  
de codes correspondant à ce texte. Des exemples sont  
donnés à la figure 5(b), dans laquelle le code de texte  
20 est {001415}, les indicateurs de texte sont conservés  
et la marque de traduction reste également dans la colonne  
de commentaires. Il convient de noter que les lignes  
vides sont traitées par le processeur comme un texte  
et reçoivent un code de texte.

25 Si on examine maintenant la figure 4(b), on  
peut voir que toutes les lignes ont été vérifiées de  
cette manière et, à l'étape 70, une liste de tous les  
codes de texte est préparée. A l'étape 71, le texte  
correspondant à chaque code de texte est affiché avec  
30 d'autres données, qui seront affichées avec celles-ci  
sur le même écran. Un traducteur introduit alors une  
traduction dans une langue particulière et, à l'étape  
72, le processeur établit des références croisées du  
code de texte avec un emplacement de mise en mémoire  
35 dans une base de données de traduction pour cette langue.

1 La base de données de traduction est indiquée par le  
numéro 73. Ces étapes sont répétées pour chaque ligne  
de code de source général, jusqu'à la production d'une  
base de données de traduction complète. L'opération est  
5 répétée à nouveau pour chaque langue, si bien qu'on  
obtient une base de données de traduction pour chaque  
langue souhaitée.

A l'étape 74, une ligne de code de source  
général est introduite dans la mémoire et, à l'étape 75,  
10 elle est vérifiée pour s'assurer qu'il existe un code  
de texte inséré dans la ligne. Si c'est le cas, le code  
de texte est extrait à l'étape 76 et est utilisé à l'étape  
77 pour reprendre le texte correspondant dans la base  
de données de traduction 73. A l'étape 78, le processeur  
15 enregistre automatiquement le texte traduit repris dans  
la base de données de traduction 73 dans la ligne de  
code de source générale, afin de produire une ligne de  
code de source finale qui est mise en mémoire à l'étape  
79. Des exemples de lignes de code de source finales  
20 sont donnés à la figure 5(c), dans laquelle il convient  
de constater que le code de texte a été remplacé par  
un texte provenant d'une base de données en langue  
allemande. Les lignes de code de source générales qui  
ne comprennent pas un code de texte sont également mises  
25 en mémoire à l'étape 79 sous forme de lignes de code  
de source finales. Suivant la vérification effectuée  
à l'étape 80, ce processus est répété pour chaque ligne  
de code de source générale. Enfin, l'étape 10 de la figure  
4(b) constitue l'étape de compilation de toutes les lignes  
30 de code de source finales.

On comprendra que la production de modules  
de code de source effectuée de cette manière avec le  
texte dans la langue souhaitée permet de produire un  
système de bases de données pour de nombreuses langues  
35 différentes, et cela d'une manière simple, économique,

1 précise et cohérente. Il faut assez bien de travail pour  
produire les bases de données de traduction mais, une  
fois que ceci a été effectué, des systèmes de bases de  
données pour différents pays peuvent être préparés très  
5 rapidement. Le texte qui est affiché en utilisant le  
système de bases de données est généralement clair et  
s'accorde bien aux autres données affichées, et il existe  
une cohérence complète entre les traductions des différents  
éléments du texte. Ce qui a été réalisé grâce à la  
10 présente invention est l'optimisation du travail humain  
pour le texte de traduction conformément au contexte  
d'un écran ainsi que l'élimination simultanée des  
incohérences entre les traductions, qui se produiraient  
sinon dans le cas où le code de texte ne serait pas  
15 produit automatiquement.

On comprendra donc que l'invention permet la  
préparation d'un système de bases de données d'une manière  
efficace, celui-ci étant généralement exempt d'erreurs  
et permettant une détection facile des erreurs.

20 L'invention n'est pas limitée aux modes de  
réalisation décrits ci-dessus mais peut subir des  
variations de construction et de détail.

1

## R E V E N D I C A T I O N S

5

1. Procédé de production et de fonctionnement  
d'un système de bases de données qui comprend les étapes  
suivantes :

détermination des impératifs propres au système  
de bases de données;

10

conception d'une structure de mise en mémoire de  
la base de données et utilisation de cette conception  
pour produire une structure de mise en mémoire de  
bases de données de référence en effectuant les  
opérations suivantes :

15

produire et mettre en mémoire une structure  
de cellules de mémoire vierges pour chacune  
des différentes zones de données et mise en  
mémoire d'un indicateur indiquant quels sont  
les types de caractères autorisés pour la mise  
en mémoire des données dans les cellules;

20

produire et mettre en mémoire un fichier  
d'adresses pour les structures de zones;

25

afficher une représentation d'une structure  
de mise en mémoire de bases de données qui  
comprend plusieurs structures de zones de  
données et vérifier celle-ci après modification  
si nécessaire; et

30

mettre en mémoire la structure de base de  
données vérifiée comme structure de base de  
données de référence dans un dispositif de  
mémoire permanente;

35

1

duplication de la structure de la base de données  
de référence afin d'obtenir une structure de mise  
en mémoire de base de données active;

5

production d'une liste de fonctions de bases  
de données pour mettre en mémoire, reprendre  
et traiter les données dans la structure de  
mise en mémoire de la base de données de  
référence;

10

enregistrement des conditions autorisées  
régissant les opérations de mise en mémoire,  
de reprise et de traitement;

15

enregistrement des évènements requis pour mettre  
le système à l'arrêt en cas de défaut grave,  
tel que déterminé par l'ensemble des conditions  
autorisées;

20

production d'un code pour définir une structure  
de point de rupture pour traiter les contenus  
de l'état et de la mémoire;

25

production d'un code pour mettre en mémoire  
un fichier des données de points de rupture;

30

conformément à la liste des fonctions,  
production de plusieurs modules de code de  
source général sans texte;

35

insertion d'un texte dans la langue du pays  
de destination dans le code de source général,  
afin de produire un code de source final;

- 1 compilation du code de source final;
- intégration des modules;
- 5 vérification du code de machine; et
- après utilisation du système de bases de données, modification de celui-ci pour améliorer ses performances en utilisant le matériel et le logiciel disponibles.
- 10
2. Procédé selon la revendication 1, dans lequel l'étape de modification du système comprend les sous-étapes suivantes :
- 15
- modifier la structure de mise en mémoire de la base de données de référence;
- comparer la structure de référence modifiée à la structure active;
- 20
- produire des données de différence; et
- modifier automatiquement la structure active pour la mettre en conformité avec la structure de référence modifiée.
- 25
3. Procédé selon la revendication 1 ou 2, dans lequel l'étape de la production des modules de code de source général comprend les sous-étapes suivantes :
- 30
- insertion d'un indicateur dans le code de source indiquant la présence d'un texte;
- insertion d'une marque indiquant que le texte
- 35

- 1 ne doit pas être traduit:
- affichage séquentiel de chaque élément de texte  
à traduire;
- 5 optimisation de la disposition du texte;
- vérification de la disposition du texte;
- 10 exécution des opérations suivantes sur chacune  
des lignes du code de source ;
- vérification de la présence d'une marque de  
texte;
- 15 conformément à cette constatation, extraction  
du texte à traduire, s'il y a lieu;
- comparaison du texte avec tous les textes  
préalablement extraits;
- 20 assignation d'un nouveau code de texte si celui-  
ci n'a pas déjà été extrait;
- 25 reprise d'un code de texte associé s'il a déjà  
été extrait; et
- insertion du code de texte repris et assigné  
dans la ligne de code de source.
- 30
4. Procédé selon la revendication 3, dans lequel  
l'étape de l'insertion d'un texte dans le module de code  
de source général comprend les sous-étapes suivantes :

1

production d'une base de données de traduction  
ayant un enregistrement de texte traduit pour  
chaque code de texte, une base de données de  
traduction existant pour chaque langue;

5

production d'un ensemble de codes de machines  
pour un programme dans une langue, en exécutant  
les étapes suivantes :

10

filtrage de chaque ligne de code de source  
et extraction d'un code de texte, s'il existe;

15

utilisation du code de texte pour reprendre  
le texte traduit correspondant pour la base  
de données de traduction appropriée;

20

insertion du texte traduit dans la ligne de  
code de source à la place du code de texte;  
et

compilation du code de source.

25

5. Procédé essentiellement tel que décrit  
ci-dessus en se référant aux figures en annexe et tel  
que représenté par celles-ci.

30

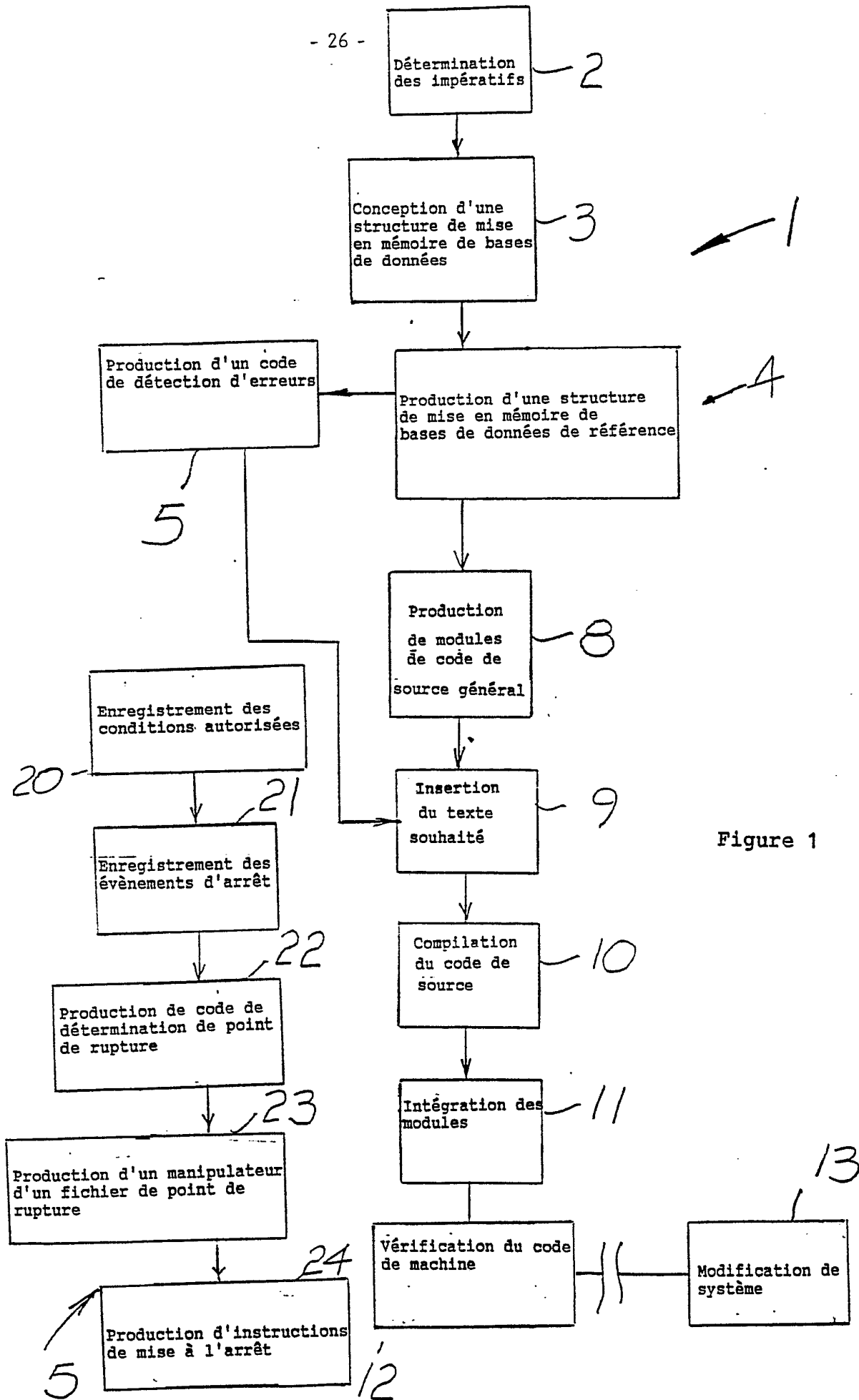


Figure 1

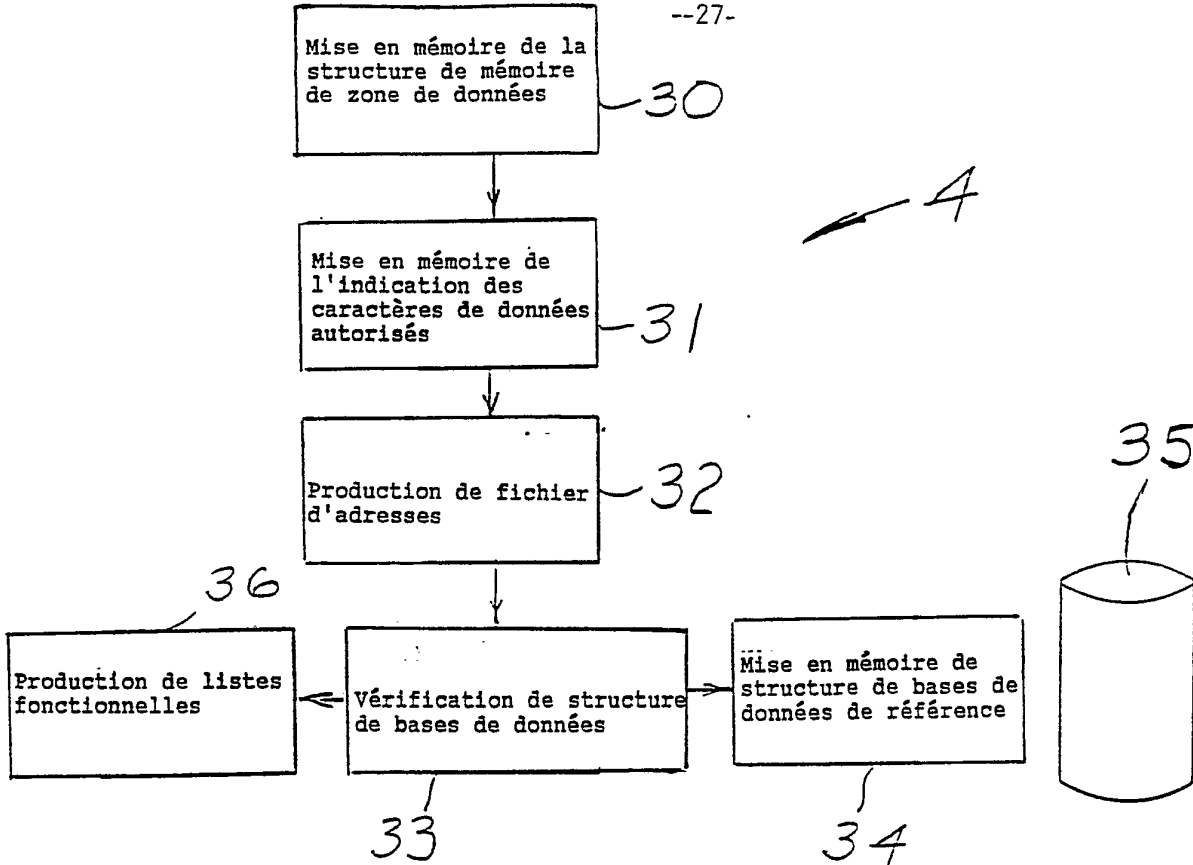


Figure 2

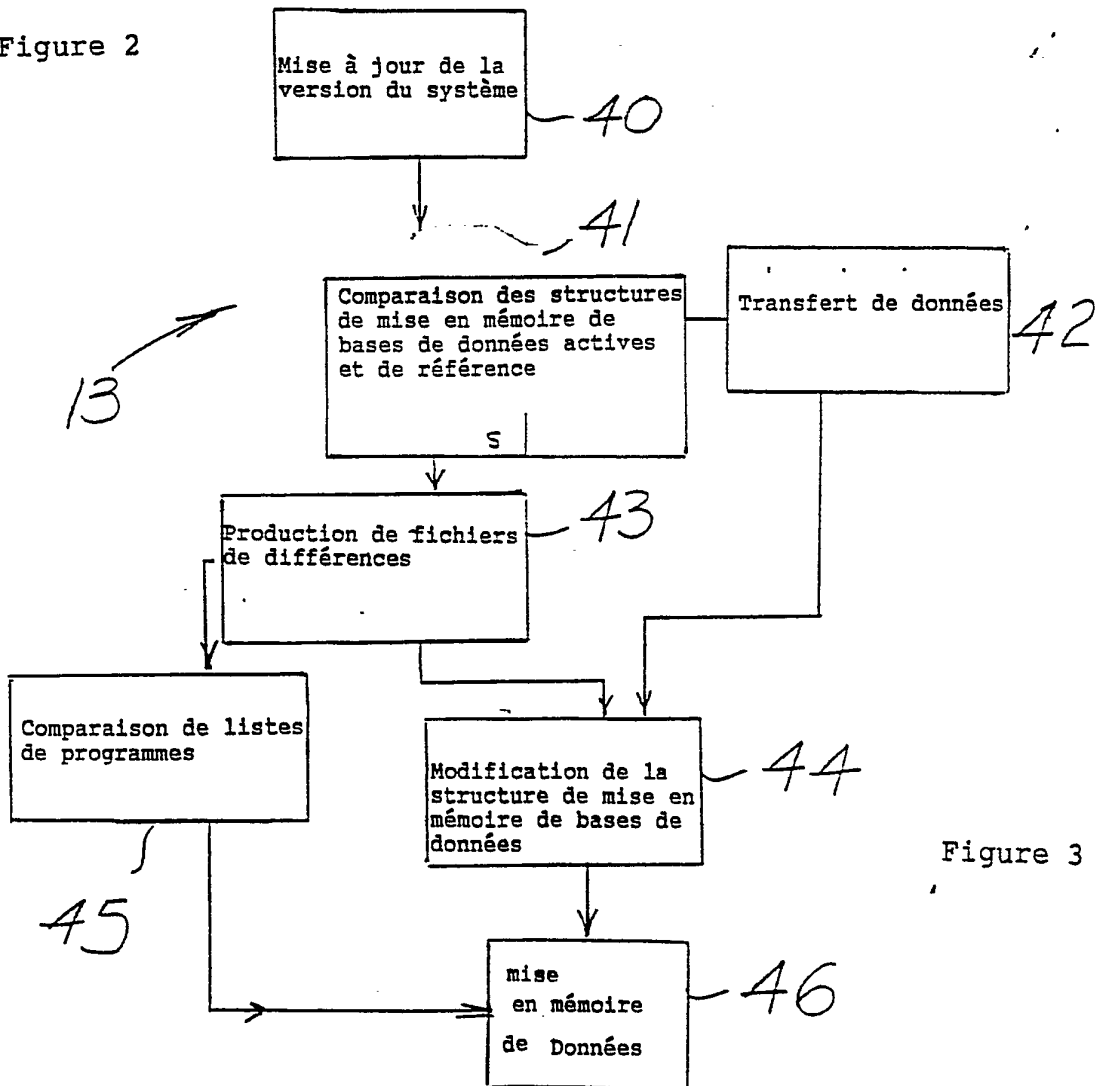


Figure 3

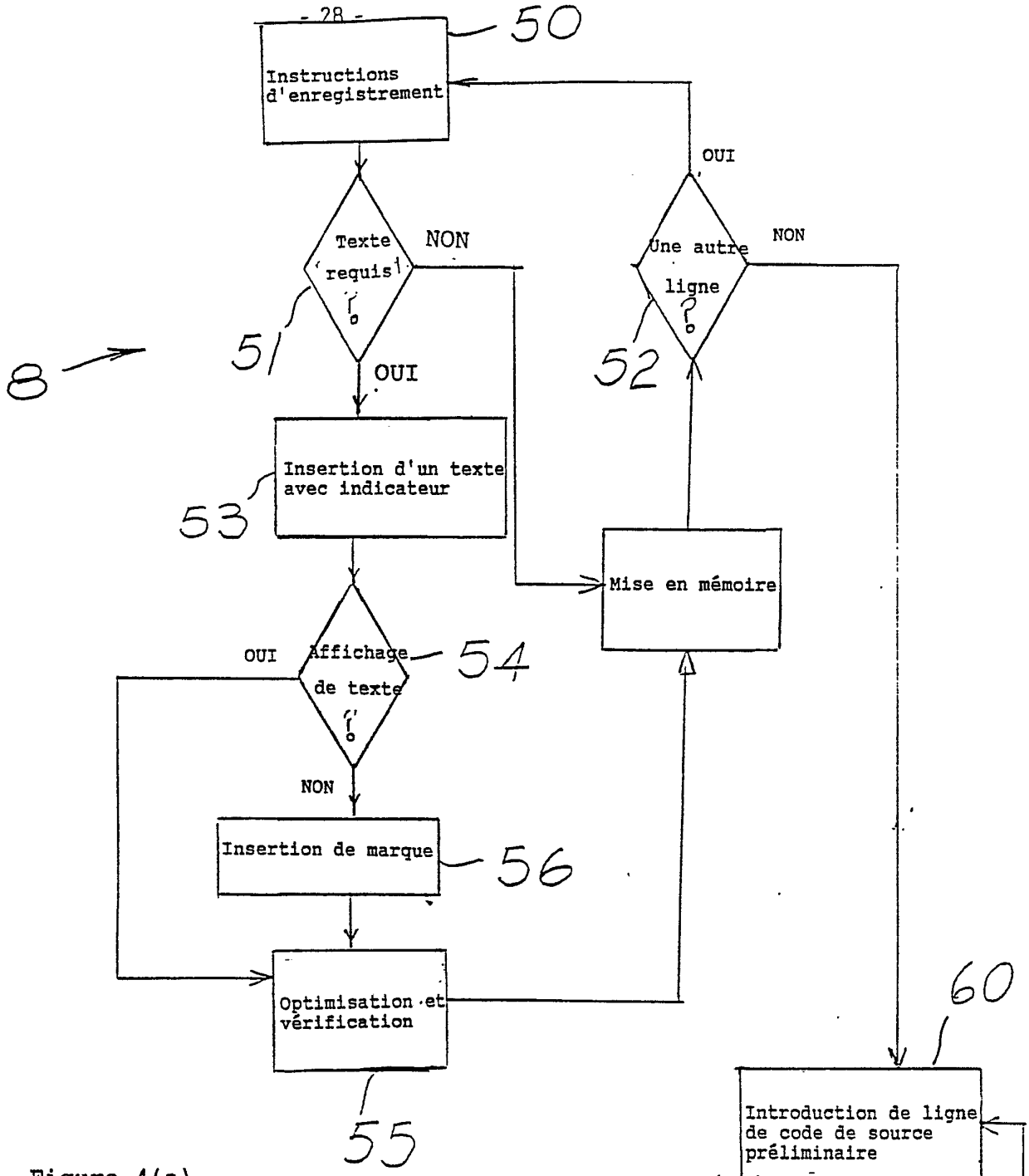
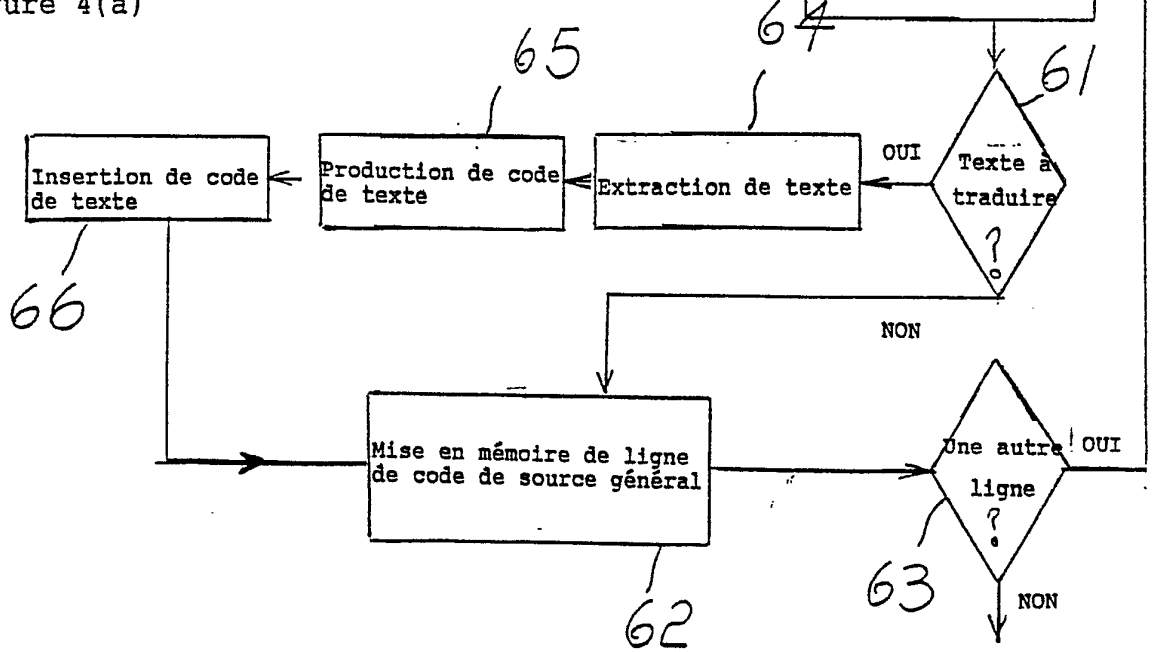


Figure 4(a)



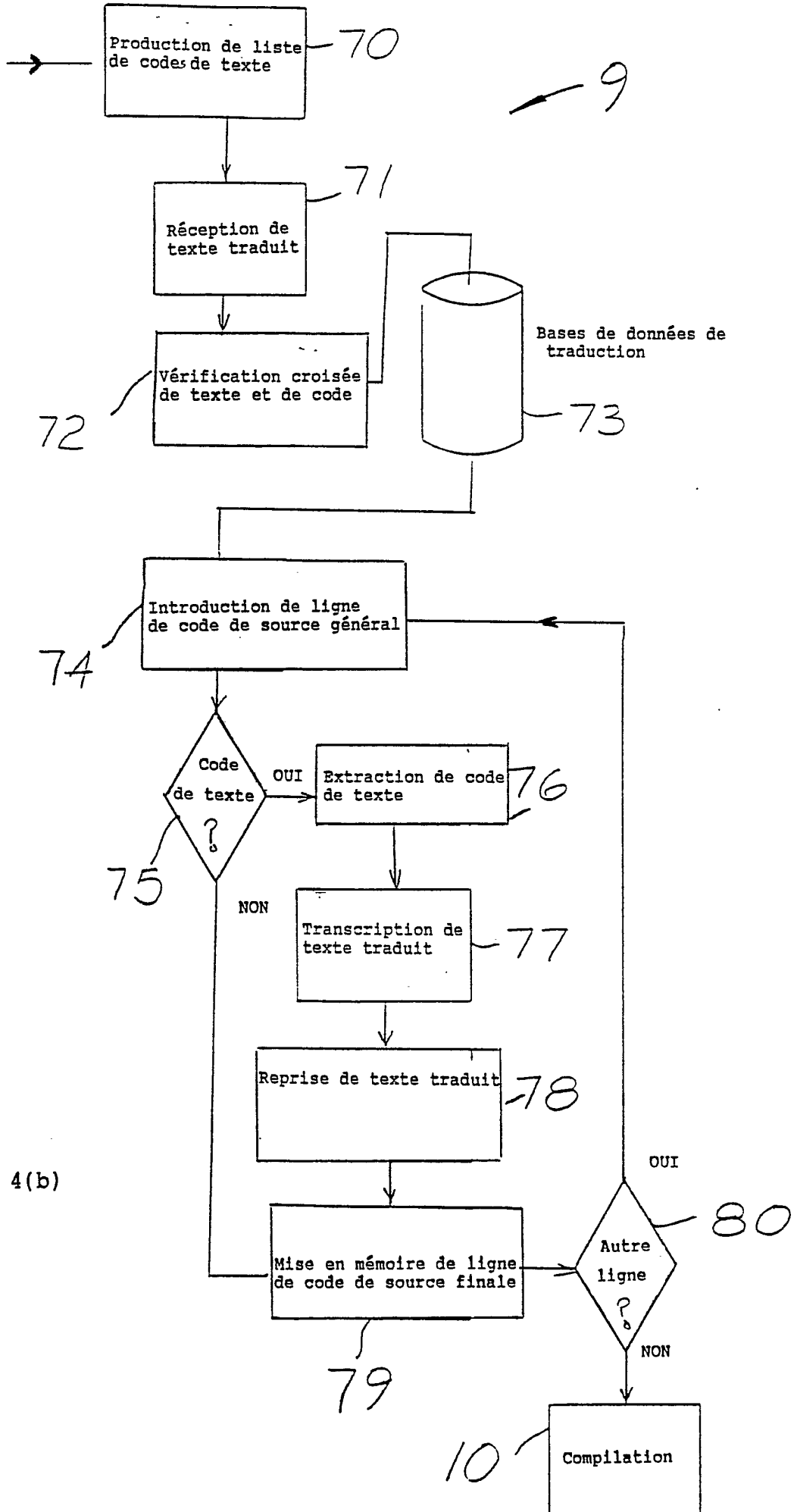


Figure 4(b)

```

@ 4,0 CLEAR TO 22,79   -30 -
@ 6,0
TEXT

```

```

End Print Command . . . .

```

Figure 5(a)

```

Description . . . . .

```

```

Printer Width . . . . . [80/132]

```

```

ENDTEXT

```

```

@ 5,0 TO 19,79 DOUBLE

```

```

@ 07,30 GET Z_Field(1) PICT '@!'      && XD_NAME [SYSTEXT]

```

```

@ 4,0 CLEAR TO 22,79

```

```

@ 6,0

```

```

TEXT

```

```

{001415}

```

```

{001415}

```

```

{001415}

```

```

{001415}

```

```

{001415}

```

```

{001415}

```

```

ENDTEXT

```

```

@ 5,0 TO 19,79 DOUBLE

```

```

@ 07,30 GET Z_Field(1) PICT '@!'      && XD_NAME [SYSTEXT]

```

Figure 5(b)

```

@ 4,0 CLEAR TO 22,79

```

```

&& Clear Text Area

```

```

@ 6,0

```

```

&& Posn cursor to just below menu.

```

```

TEXT

```

```

Druckbefehl beenden . . .

```

```

Beschreibung . . . . .

```

Figure 5(c)

```

Drucker Breite . . . . . [80/132]

```

```

ENDTEXT

```

```

@ 5,0 TO 19,79 DOUBLE

```

```

@ 07,30 GET Z_Field(1) PICT '@!'      && XD_NAME [SYSTEXT]

```