

12)

DEMANDE DE BREVET D'INVENTION

A1

22) Date de dépôt : 03.07.00.

30) Priorité :

43) Date de mise à la disposition du public de la demande : 04.01.02 Bulletin 02/01.

56) Liste des documents cités dans le rapport de recherche préliminaire : *Se reporter à la fin du présent fascicule*

60) Références à d'autres documents nationaux apparentés :

71) Demandeur(s) : AXICARE Société par actions simplifiée — FR.

72) Inventeur(s) : TREIL NICOLAS HERVE EDOUARD, GREGOR SERGE JEAN FRANCOIS et LECLERC VINCENT PHILIPPE.

73) Titulaire(s) :

74) Mandataire(s) : CABINET NETTER.

54) PROCÉDE DE TRAITEMENT DE DONNEES STRUCTUREES UTILISANT UN LANGAGE INFORMATIQUE ORIENTE OBJET.

57) Le procédé de traitement de données, comprend les étapes suivantes:

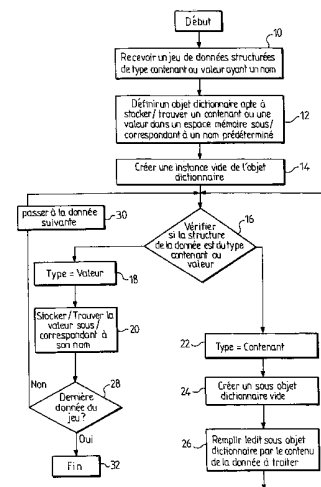
a) recevoir un jeu de données à traiter, structurées et reliées entre elles, de type valeur ayant un nom et/ ou contenant ayant un nom et aptes à contenir au moins une valeur ayant un nom et/ ou au moins un autre contenant ayant un nom;

b) définir un objet dictionnaire apte à stocker/ trouver dans un espace mémoire un contenant ou une valeur sous/ correspondant à un nom prédéterminé;

c) créer une instance vide de l'objet dictionnaire;

d) vérifier si la structure de la donnée à traiter est du type valeur ou contenant;

e) en cas de donnée de type valeur, stocker/ trouver la valeur dans l'espace mémoire sous/ correspondant au nom de la donnée à traiter, et revenir à l'étape d) jusqu'à la dernière donnée à traiter dudit jeu tandis qu'en cas de donnée de type contenant, créer un sous objet dictionnaire vide conformément à l'étape c), remplir ledit sous objet dictionnaire par le contenu de la donnée à traiter conformément aux étapes d) à e), et revenir à l'étape d) jusqu'à traiter la dernière donnée du jeu.



Procédé de traitement de données structurées utilisant un langage informatique orienté objet

5

La présente invention concerne le traitement de données structurées utilisant un langage informatique orienté objet.

10 Elle trouve une application générale dans la modélisation informatique de données, la visualisation de données et la construction d'applications informatiques client-serveur.

15 On connaît déjà des procédés de traitement de données permettant de gérer des données structurées ou hiérarchiques, dont le niveau le plus bas est par exemple un document (texte, image, son, etc.).

20 Par exemple, on connaît un procédé qui permet le lien, via un serveur d'application, entre des pages émanant ou transitant par le réseau Internet, appelées encore pages sur la toile ou « pages web », et une base de données. Dans ce genre de procédé, les champs ou variables utilisables au niveau des pages web doivent faire référence à des champs décrits dans le schéma de la base de données utilisée.

25

Un tel procédé a l'inconvénient de limiter la structuration des données à un schéma figé déterminé à l'avance. De plus, aucune capacité d'apprentissage n'est possible avec ce procédé. Il en résulte que la mise à jour du schéma des données implique une modification conjointe de la base de données et des modèles de page, ce qui alourdit, ralentit et complexifie cette mise à jour.

30

35 On connaît aussi un procédé utilisant un langage orienté objet et permettant l'échange de données entre des pages (ou formulaires) web et une base de données. Dans ce genre de procédé, la base de données est généralement de type relationnelle, ce qui exclut toute possibilité d'apprentissage. De plus, la correspondance entre les données des pages et les objets se fait en utilisant des champs définis par la classe

40

d'objet. Ainsi, une donnée qui ne correspond pas à un champ de la classe d'objet ne peut être géré dans un modèle de page.

5 La présente invention remédie à ces inconvénients.

Elle porte sur un traitement de données structurées utilisant un langage informatique orienté objet.

10 Selon une définition générale du procédé selon l'invention, le procédé comprend les étapes suivantes :

a) recevoir un jeu de données à traiter, structurées et reliées entre elles, de type valeur ayant un nom et/ou  
15 contenant ayant un nom et apte à contenir au moins une valeur ayant un nom et/ou au moins un autre contenant ayant un nom;

b) définir un objet dictionnaire apte à stocker/trouver dans un espace mémoire un contenant ou une valeur sous/correspondant à un nom prédéterminé ;  
20

c) créer une instance vide de l'objet dictionnaire ;

d) vérifier si la structure de la donnée à traiter est du  
25 type valeur ou contenant ;

e) en cas de donnée de type valeur, stocker/trouver la valeur dans l'espace mémoire sous/correspondant au nom de la donnée à traiter, et revenir à l'étape d) jusqu'à traiter la  
30 dernière donnée dudit jeu ;

f) tandis qu'en cas de donnée de type contenant, créer un sous objet dictionnaire vide conformément à l'étape c), remplir ledit sous objet dictionnaire par le contenu de la  
35 donnée à traiter conformément aux étapes d) à f), et revenir à l'étape d) jusqu'à traiter la dernière donnée du jeu.

L'apport du procédé selon l'invention est de permettre une génération d'applications extrêmement rapides et une structure des données orientées objets totalement souple.

5 Par exemple, lorsque le procédé selon l'invention est appliqué au stockage de données dans une base de données, le procédé selon l'invention comprend les étapes suivantes :

1) prévoir une base de données ;

10

2) recevoir un jeu de données rangées selon une structure hiérarchique du type valeur ayant un nom et/ou contenant ayant un nom et apte à contenir au moins une valeur et/ou au moins un autre contenant ayant un nom ;

15

3) prévoir une instance de base de donnée remplie d'un objet dictionnaire ;

4) en cas de donnée de type valeur, stocker ou remplacer la  
20 valeur sous son nom dans l'objet dictionnaire ;

5) en cas de donnée de type contenant, vérifier s'il existe un objet dictionnaire sous le nom de la donnée à traiter ;

25 6) dans l'affirmative, remplir l'objet dictionnaire par le contenu de la donnée et revenir à l'étape 4) jusqu'à traiter la dernière donnée du jeu tandis que dans la négative, créer un sous-objet dictionnaire vide et remplir le sous-objet dictionnaire par le contenu de la donnée à traiter et revenir  
30 à l'étape 4) jusqu'à traiter la dernière donnée du jeu.

En pratique, les données à traiter sont reçues sous la forme d'objet dictionnaire afin d'être rangées en série les unes à la suite des autres dans un espace de stockage choisi. Dans  
35 ce cas, le procédé selon l'invention comprend les étapes suivantes :

I) recevoir un objet dictionnaire rempli d'au moins une donnée à stocker dans un espace de stockage choisi ;

II) pour chaque nom de valeur et/ou contenant stocké dans l'objet dictionnaire, analyser la structure des données contenues dans l'objet dictionnaire sous ce nom ;

5 III) en cas de donnée de type valeur, écrire dans l'espace de stockage choisi le nom de la valeur suivi d'un marqueur de valeur et de la valeur elle-même tandis qu'en cas de donnée de type contenant, écrire, dans l'espace de stockage choisi, un marqueur de contenant, le nom du contenant, un marqueur de  
10 valeur, et le contenu de la donnée à traiter et un marqueur de fin de contenant.

Selon un autre mode préféré de mise en oeuvre du procédé selon l'invention, l'objet dictionnaire est capable de stocker en  
15 outre au moins une méthode de calcul sous un nom prédéterminé, et à trouver comme valeur sous ce nom, le résultat de la méthode de calcul associée.

Par exemple, les données à traiter émanent d'un formulaire ou  
20 page dans lequel les données sont structurées et reliées entre elles avec des données de type valeur ayant un nom et/ou contenant ayant un nom et apte à contenir une valeur ou un contenant ayant un nom.

25 D'autres caractéristiques et avantages de l'invention apparaîtront à la lumière de la description détaillée ci-après des dessins dans lesquels :

- la figure 1 est un organigramme illustrant les étapes  
30 générales du procédé selon l'invention ;

- la figure 2 est un organigramme illustrant le procédé de l'invention dans la phase stockage de données dans une base de données ; et

35

- la figure 3 est un organigramme illustrant le procédé de persistance d'une base de données selon l'invention.

La présente invention décrit un mécanisme générique d'encapsulation de données adapté à toute structure de données. Le mécanisme d'encapsulation selon l'invention utilise un objet de base appelé ci-après objet dictionnaire.

5

On entend ici par objet dictionnaire, un objet qui permet de traiter (stocker et/ou trouver) une donnée de type valeur ayant un nom ou contenant ayant un nom, (le contenant étant apte à contenir lui même au moins une valeur ayant un nom ou  
10 au moins un autre contenant ayant un nom).

La fonction de base d'un objet dictionnaire est par conséquent de stocker dans un espace mémoire un contenant ou une valeur sous un nom prédéterminé et aussi de trouver dans un  
15 espace mémoire ledit contenant ou ladite valeur correspondant à un nom prédéterminé.

La structure d'un objet dictionnaire est simple. La classe d'objets dictionnaires contient un tableau des noms des  
20 contenants et des valeurs. La classe d'objets dictionnaires comprend aussi un tableau des contenants et des valeurs étiquetés, c'est-à-dire rangés selon un index correspondant à celui des noms desdits contenants ou valeurs.

25 Par exemple, les techniques dites de hachage, telles que celles appelées "hashtable" en langage JAVA, ou celles décrites dans l'ouvrage "The art of computer programming, Vol 1-3 boxed set HBK, de Donald E. KNUTH, 1999", permettent de réaliser des objets dictionnaires.

30

Dans l'art antérieur, la plupart des procédés d'encapsulation de données supposent que les objets possèdent une structure prédéfinie et qu'on dispose d'une connaissance précise de la structure des données. Le procédé d'encapsulation est  
35 généralement capable de traiter les données dont la structure correspond à celle de l'objet. Par exemple, une donnée ayant une structure nom, prénom, et date de naissance, est convenablement traitée si l'objet « personne » a pour variables nom, prénom, et date de naissance. Par contre, la donnée

"adresse" n'est pas traitée si la structure de l'objet « personne » ne comprend pas ce type de variable.

5 Le procédé selon l'invention permet justement d'encapsuler des données dont la structure n'est pas connue a priori.

Dans l'exemple mentionné ci-avant, l'objet dictionnaire DIC1, selon l'invention, stocke la valeur "Dupont" dans la variable ayant pour nom "nom", stocke la valeur "Paul" dans la variable ayant pour nom "prénom" et stocke un objet dictionnaire DIC2 sous le nom "adresse".

15 Selon l'invention, l'objet dictionnaire DIC2 est alors capable de stocker la valeur "3, allée des Violettes" dans la variable ayant pour nom "rue" ; stocke la valeur "Paris" sous le nom "ville" ; et stocke la valeur "75012" sous le nom "code-postal".

20 En référence à la figure 1, le procédé de traitement de données selon l'invention peut être résumé sous l'organigramme suivant :

- recevoir un jeu de données structurées selon une hiérarchie et reliées entre elles, la structure étant du type valeur ayant un nom et/ou contenant ayant un nom et apte à contenir une valeur et/ou un autre contenant ayant un nom (étape 10);

30 - définir un objet dictionnaire apte à stocker/ trouver un contenant ou une valeur dans un espace mémoire sous/correspondant à un nom prédéterminé (étape 12) ;

35 - créer une instance vide de l'objet dictionnaire (étape 14);

- vérifier si la structure de la donnée à traiter est du type contenant ou valeur (étape 16);

- en cas de donnée de type valeur (étape 18), stocker/-  
trouver la valeur sous son nom (étape 20) tandis qu'en cas de  
donnée de type contenant (étape 22), créer un sous-objet  
dictionnaire vide (étape 24) conformément à l'étape 14,  
5 remplir ledit sous-objet dictionnaire par le contenu de la  
donnée (étape 26) conformément aux étapes 16 à 32, et revenir  
à l'étape 16 jusqu'à traiter la dernière donnée dudit jeu  
(étapes 28, 30 et 32) .

10 Avantageusement, le procédé selon l'invention permet en outre  
d'ajouter aux données d'autres variables dont les valeurs ne  
sont pas présentes dans les objets ou contenants.

Ainsi dans l'exemple mentionné ci-avant, l'objet ou contenant  
15 "personne" peut stocker la valeur "date de naissance" et être  
encapsulé dans un contenant "personne" qui est capable  
d'afficher en outre l'âge selon une méthode de calcul  
appropriée.

20 Avantageusement, l'invention permet de traiter ces variables  
dont les valeurs sont issues de méthodes de calcul. Ces  
valeurs sont avantageusement rangées selon l'invention au  
moyen d'un objet dictionnaire permettant d'associer le nom  
d'une valeur et la méthode de calcul utilisée pour fournir la  
25 valeur.

Plus précisément, après réception d'un nom de valeur, il est  
vérifié si ce nom de valeur est associé à une méthode de  
calcul dans l'objet dictionnaire. En présence d'une méthode  
30 associée, il est prévu de calculer la valeur selon la méthode  
de calcul associée et d'obtenir ainsi un résultat. Dans la  
négative, il est prévu de rechercher dans l'objet diction-  
naire des données formant résultat et de délivrer le résultat  
correspondant.

35 Le procédé selon l'invention trouve plusieurs applications.  
L'une d'elles consiste à introduire dans une base de données  
de nouvelles données dont la structure n'est pas connue a  
priori.

Par exemple, les données sont saisies dans des pages d'interface utilisateur, appelées formulaires ou pages de saisie.

La présente invention propose un formalisme permettant de  
5 nommer chaque champ ou variable du formulaire de façon à  
générer des données auto-structurées pouvant être persistées  
(mises à jour) dans un espace de stockage choisi (base de  
données par exemple).

10 Le formalisme consiste par exemple à nommer chaque champ du  
formulaire de la façon suivante :

nom d'objet / nom de sous-objet 1 /... / nom de sous-objet n  
/ nom de valeur .

15

Les objets et sous-objets sont ici des contenants ayant un  
nom.

Il est à remarquer que la barre oblique / est un séparateur  
20 arbitraire qui peut être remplacé par tout autre caractère  
arbitraire.

On peut aussi prévoir des marqueurs de fin de contenant et/ou  
de fin de valeur.

25

certaines variables (contenant et/ou valeur) peuvent être  
vides.

Ce formalisme permet, lorsque les données et noms de champs  
30 sont envoyés à un serveur d'application, de générer des  
objets (contenants) correspondants à la structure de données  
désirée.

Par exemple, un formulaire comportant les champs et valeurs  
35 suivants :

-	PERSONNE/NOM	DUPONT
-	PERSONNE/PRÉNOM	Jean

- PERSONNE/VOITURE/MARQUE Ford
- PERSONNE/VOITURE/MODELE Escort

5 Un tel formulaire permet de générer l'objet suivant :

Objet dictionnaire Dic 1, de type dictionnaire stocke la valeur "Dupont" sous le nom de champ "nom", stocke la valeur "Jean" sous le nom de champ "prénom" et stocke un objet  
10 dictionnaire, Dic 2, de type dictionnaire, sous le nom de champ "voiture".

L'objet dictionnaire DIC2, de type dictionnaire stocke alors la valeur "Ford" sous le nom de champ "marque" et aussi la  
15 valeur "Escort" sous le nom de champ "modèle".

On a représenté en figure 2 un organigramme qui illustre la procédure de stockage d'une valeur selon l'invention.

20 Selon l'étape 100, on prévoit ou on reçoit un objet dictionnaire DIC1, un nom de champ (étape 102) et la valeur attribuée correspondante (étape 104).

A l'issue des étapes 100, 102 et 104, on peut passer au stockage de la valeur (étape 106).  
25

Pour chaque nom de champ, on analyse la structure des données contenues dans ledit champ sous ce nom (étape 108). En pratique, on vérifie si le nom de champ contient la barre oblique « / ».  
30

Si le nom de champ ne contient pas le caractère séparateur arbitraire, (ici barre oblique), on stocke la valeur sous le nom « Nom de champ » dans l'objet dictionnaire (étape 110).  
35 En effet, à ce stade, la donnée est du type variable simple, c'est-à-dire du type donnée contenant seulement une valeur sous un nom.

En revanche, si le nom de champ contient le séparateur arbitraire, cela signifie que la donnée est du type « contenant » propre à contenir un autre objet ou sous-objet. On extrait alors le nom de ce champ (étape 112) et on vérifie si un sous-objet existe dans l'objet dictionnaire sous ce nom (étape 114).

Dans l'affirmative, on obtient le sous-objet du dictionnaire (étape 116) et on modifie le nom de champ (étape 124) en supprimant la partie précédant le séparateur "/", ainsi que le séparateur lui-même (étape 118).

En revanche, s'il n'existe pas de sous-objet dans l'objet dictionnaire, on prévoit de créer un sous-objet (étape 120) et on le stocke dans l'objet dictionnaire.

Lorsque le sous-objet est ainsi créé (étape 122), il est prévu de modifier le nom de champ (étape 124) en supprimant la partie précédant le séparateur "/", ainsi que le séparateur lui-même (conformément à l'étape 118 mentionnée ci-avant).

A l'issue du procédé, on dispose ainsi d'un sous-objet (étape 126), d'un nom de champ modifié (étape 124) et d'une valeur correspondante (étape 128), ce qui permet de stocker la valeur (étape 130), par itération du procédé conformément aux étapes 108 à 130.

Si l'on utilise l'exemple mentionné ci-avant à l'organigramme de la figure 2, on dispose d'un objet dictionnaire correspondant à la description d'une personne avec des noms de champ "NOM", "PRENOM" et des valeurs "DUPONT" et "JEAN".

Cet objet dictionnaire comprend également un contenant "VOITURE" stockant la valeur "FORD" sous le nom de champ "MARQUE".

Selon les étapes 108 à 110, la valeur "JEAN" est stockée dans le nom de champ "PRENOM".

Selon les étapes 108, 112, 114, 116, 118, 124, 126, 128, 130, puis 108 à 110, la valeur "ESCORT" est stockée dans le nom de champ "VOITURE/MODELE".

- 5 Enfin, selon les étapes 108, 112, 114, 120, 122, 118, 124, 126, 128, 130, puis 108 à 110, la valeur "PARIS" est stocké dans le nom de champ "ADRESSE/VILLE".

10 En référence à la figure 3, la présente invention a également pour objet un procédé de persistance (mise à jour) des données d'un objet dictionnaire encapsulé comme décrit précédemment.

15 A partir d'un objet dictionnaire dit "logiciel", c'est-à-dire résultant d'une encapsulation telle que celle décrite précédemment (étape 200) et d'un objet base de données, dit "statique", c'est-à-dire non encapsulé (étape 202), la procédure « persister objet logiciel » est prête à débiter (étape 204).

20

Le procédé prévoit d'analyser le contenu (ici chaque nom) de l'objet dictionnaire à traiter (étapes 206 et 208).

25 En présence d'un nom, on passe à l'étape de traitement suivante (étape 212), sinon on met fin au procédé (étape 210)

30 Si le nom correspond à une variable simple, la valeur de la variable est persistée (mise à jour) dans la base de données (étape 214), conformément à des méthodes de stockage qui ne font pas l'objet de la présente invention.

35 En revanche, si la donnée est du type « contenant », on vérifie s'il y a un sous-objet statique dans la base de données possédant ce nom (étape 216).

Dans l'affirmative, il est recherché dans la base de données, un sous-objet statique correspondant à ce nom (étape 218), tandis qu'en l'absence de sous-objet statique dans la base de

données sous ce nom, il est prévu de créer un sous-objet statique avec ce nom dans la base de données (étape 220).

5 A l'issue des étapes 218 ou 220, on dispose d'un sous-objet statique, et on obtient le sous-objet logiciel de l'objet dictionnaire correspondant (étape 224), ledit sous-objet étant encapsulé conformément au procédé d'encapsulation selon l'invention.

10 A l'issue de l'étape 224, on dispose ainsi d'un sous objet logiciel (étape 226), homologue de l'objet logiciel défini à l'étape 200, et d'un sous objet statique (étape 228), issu de la base de données, homologue de l'objet statique défini à l'étape 202.

15 Avec ces sous objets statique et logiciel, il est possible de mettre à jour (persister) les données de l'objet logiciel (ici dictionnaire) (étape 230) dans l'objet statique (ici la base de données).

20 La présente invention est un procédé de génération d'application susceptible d'être implanté en utilisant un formalisme de représentation de données hiérarchiques et utilisant un langage informatique orienté objet de type Java, C++ ou  
25 analogue.

L'apport principal de la présente invention réside dans la capacité de modifier dynamiquement le schéma de description des données, en particulier de gérer facilement de nouveaux  
30 types d'objets ou de nouvelles valeurs.

De plus, dans le cas d'une application client-serveur, dans laquelle les clients interagissent avec les données (création, modification, destruction), l'invention permet de  
35 référencer des objets de la base de données dans des pages sur la toile constituant l'interface utilisateur. Elle permet ainsi de référencer des valeurs ou des contenants de valeurs non existants dans des formulaires de saisie. Une fois ces formules envoyés au serveur d'application selon l'invention,

ces nouveaux contenants et valeurs sont créés dans la base de données.

5 Une application avantageuse, mais non limitative de la présente invention se trouve dans le domaine médical, notamment lors de la mise en place de réseau de soins permettant de gérer des dossiers patients cliniques.

### Revendications

1. Procédé de traitement de données, comprenant les étapes  
5 suivantes :
- a) recevoir un jeu de données à traiter, structurées et  
reliées entre elles, de type valeur ayant un nom et/ou  
contenant ayant un nom et aptes à contenir au moins une  
valeur ayant un nom et/ou au moins un autre contenant ayant  
10 un nom ;
  - b) définir un objet dictionnaire apte à stocker/trouver  
dans un espace mémoire un contenant ou une valeur sous/cor-  
respondant à un nom prédéterminé ;
  - c) créer une instance vide de l'objet dictionnaire ;
  - 15 d) vérifier si la structure de la donnée à traiter est du  
type valeur ou contenant ;
  - e) en cas de donnée de type valeur, stocker/trouver la  
valeur dans l'espace mémoire sous/correspondant au nom de la  
donnée à traiter, et revenir à l'étape d) jusqu'à la dernière  
20 donnée à traiter dudit jeu tandis qu'en cas de donnée de type  
contenant, créer un sous objet dictionnaire vide conformément  
à l'étape c), remplir ledit sous objet dictionnaire par le  
contenu de la donnée à traiter conformément aux étapes d) à  
e), et revenir à l'étape d) jusqu'à traiter la dernière  
25 donnée du jeu.
2. Procédé selon la revendication 1, appliqué dans le  
stockage de données dans une base de données, caractérisé en  
ce qu'il comprend les étapes suivantes :
- 30
- 1) prévoir une base de données propre à contenir un jeu de  
données rangées selon une structure hiérarchique du type  
valeur ayant un nom et/ou contenant ayant un nom et apte à  
contenir au moins une valeur et/ou au moins un autre conte-  
nant ayant un nom, les données étant séparées les unes des  
35 autres par des marqueurs choisis ;
  - 2) recevoir une donnée à traiter de type valeur ou contenant  
ayant un nom ;

3) prévoir une instance remplie d'un objet dictionnaire ;

4) en cas de donnée de type valeur, stocker ou remplacer la valeur sous son nom dans l'objet dictionnaire ;

5

5) en cas de donnée de type contenant, vérifier s'il existe un objet dictionnaire sous ce nom ;

6) dans l'affirmative, remplir l'objet dictionnaire par le contenu de la donnée et revenir à l'étape 4) jusqu'à traiter la dernière donnée du jeu tandis que dans la négative, créer un sous-objet dictionnaire vide conformément à l'étape 3) et remplir le sous-objet dictionnaire par le contenu de la donnée conformément aux étapes 4 à 6, et revenir à l'étape 4) jusqu'à traiter la dernière donnée du jeu.

3. Procédé selon la revendication 1, dans lequel données à traiter sont reçues sous la forme d'un objet dictionnaire afin d'être rangées en série les unes à la suite des autres dans un espace de stockage choisi, caractérisé en ce qu'il comprend en outre les étapes suivantes :

I) prévoir un objet dictionnaire rempli de données à traiter;

II) pour chaque nom de valeur et/ou contenant stocké dans l'objet dictionnaire, analyser la structure des données contenues dans l'objet dictionnaire sous ce nom ;

III) en cas de donnée de type valeur, écrire dans l'espace de stockage choisi le nom de la valeur suivi d'un marqueur de valeur et de la valeur elle-même tandis qu'en cas de donnée de type contenant, écrire, dans l'espace de stockage choisi, un marqueur de contenant, le nom du contenant, un marqueur de valeur, et le contenu de la donnée à traiter et un marqueur de fin de contenant.

4. Procédé selon l'une des revendications précédentes, caractérisé en ce que l'objet dictionnaire est de type objet logiciel capable de stocker en outre au moins une méthode de

calcul sous un nom prédéterminé, et à trouver comme valeur sous ce nom, le résultat de cette méthode de calcul.

5. Procédé selon l'une quelconque des précédentes revendications, caractérisé en ce que les données à traiter émanent d'une page sur la toile dans laquelle les données sont structurées et reliées entre elles avec des données de type valeur ayant un nom et/ou contenant ayant un nom et apte à contenir une valeur ou un contenant ayant un nom.

1/3

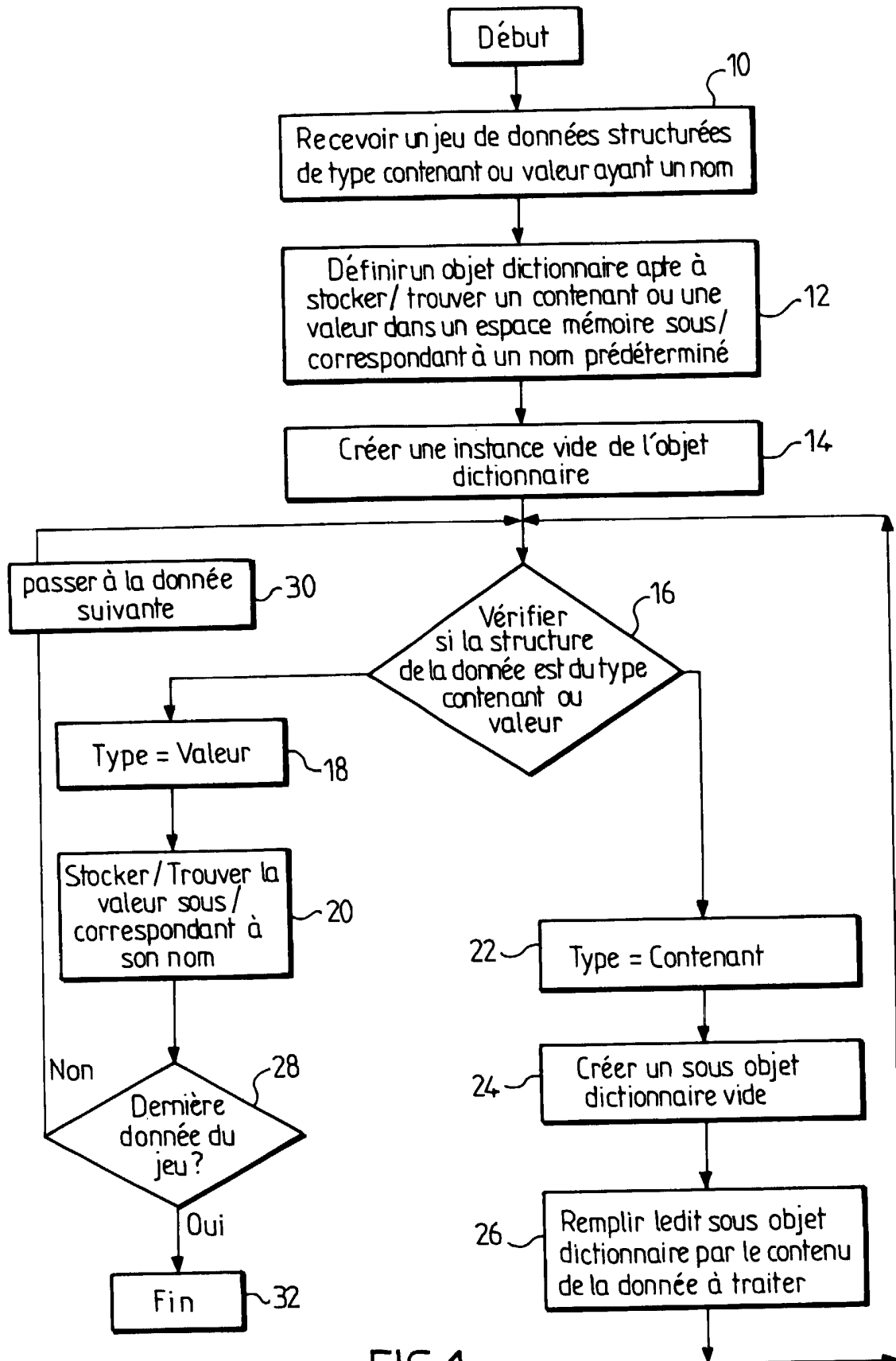


FIG. 1

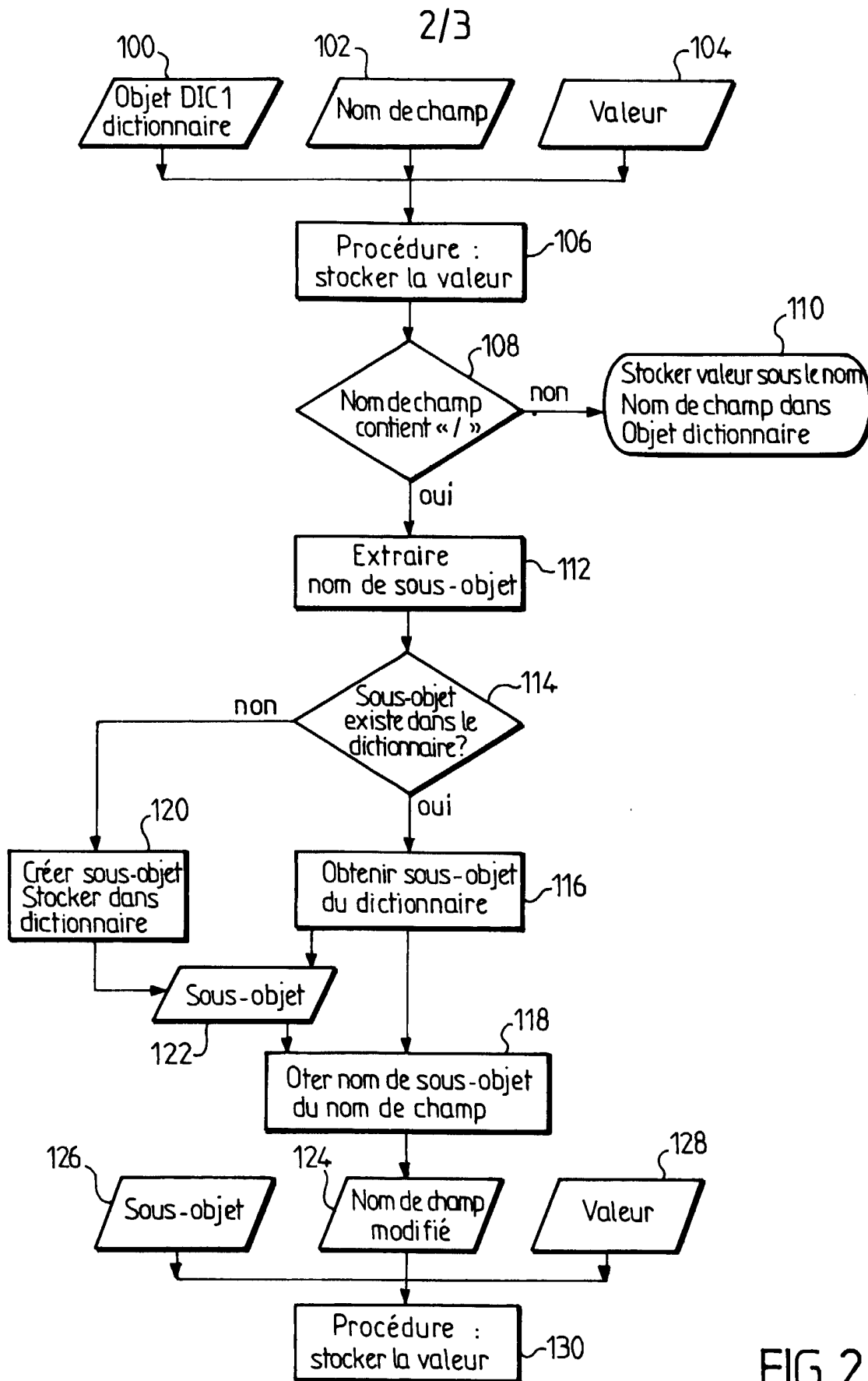


FIG. 2

3/3

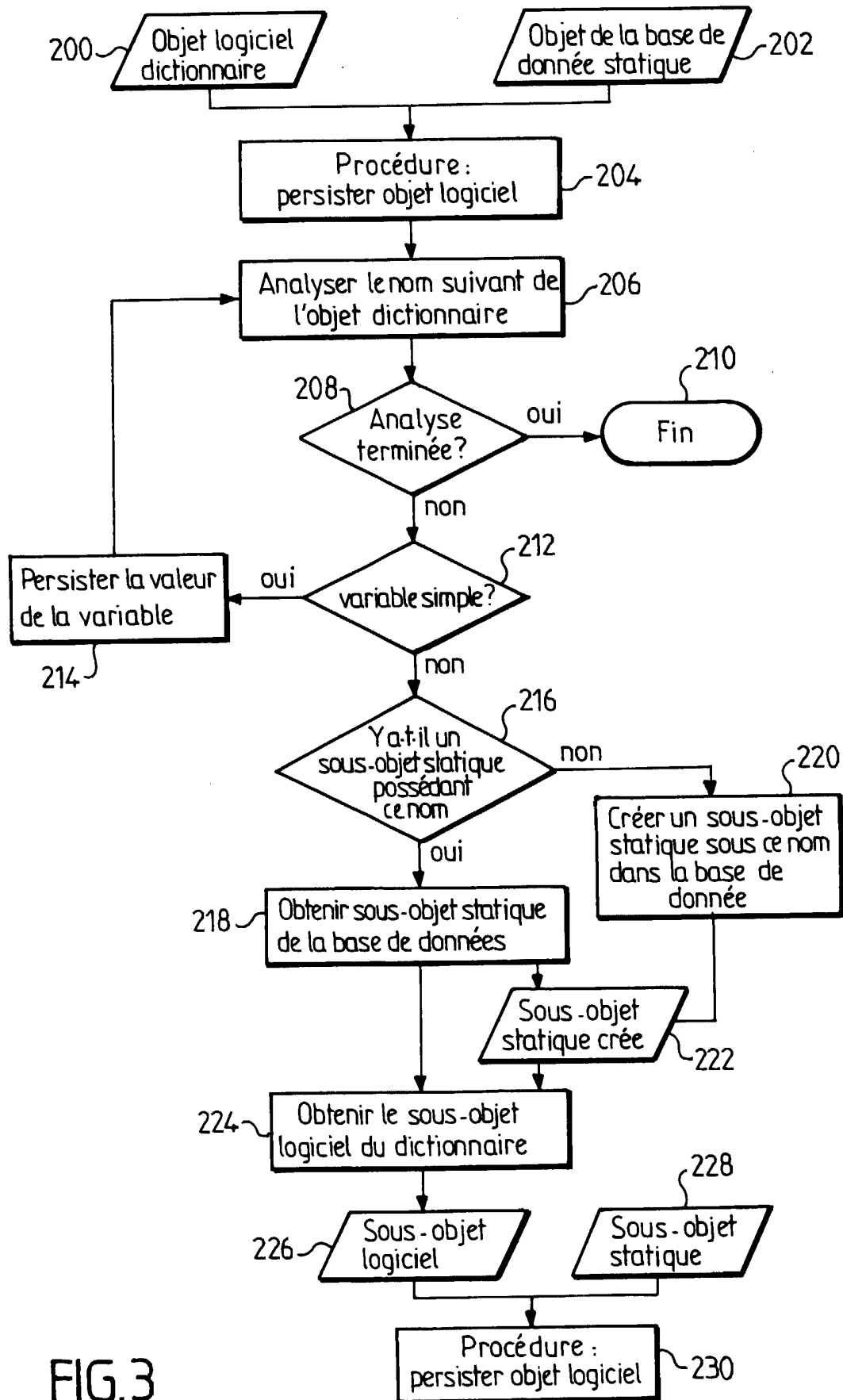


FIG.3

