

(12) DEMANDE INTERNATIONALE PUBLIÉE EN VERTU DU TRAITÉ DE COOPÉRATION EN MATIÈRE DE BREVETS (PCT)

(19) Organisation Mondiale de la  
Propriété Intellectuelle  
Bureau international



(10) Numéro de publication internationale

WO 2014/090514 A1

(43) Date de la publication internationale  
19 juin 2014 (19.06.2014)

WIPO | PCT

- (51) Classification internationale des brevets :  
G06F 9/44 (2006.01) G06F 17/50 (2006.01)
- (21) Numéro de la demande internationale :  
PCT/EP2013/074093
- (22) Date de dépôt international :  
18 novembre 2013 (18.11.2013)
- (25) Langue de dépôt : français
- (26) Langue de publication : français
- (30) Données relatives à la priorité :  
1261871 11 décembre 2012 (11.12.2012) FR
- (71) Déposant : TECHFORM [FR/FR]; Immeuble Diapason  
Bât A, Rue Jean Bart, BP 57481, F-31674 Labège (FR).
- (72) Inventeurs : BENSOUSSAN, Charles; 5 rue Ella  
Maillart, F-31300 Toulouse (FR). BENSOUSSAN, Alain;  
387 Chemin de Saint Papoul, F-31450 Odars (FR).
- (74) Mandataire : CORNUEJOLS, Christophe; SCHMIT-  
CHRETIEN, Parc de Basso Cambo, 4, rue Paul Mesple, F-  
31100 Toulouse (FR).
- (81) États désignés (sauf indication contraire, pour tout titre  
de protection nationale disponible) : AE, AG, AL, AM,  
AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,  
BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM,  
DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,  
HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR,  
KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME,  
MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ,  
OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA,  
SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM,  
TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM,  
ZW.
- (84) États désignés (sauf indication contraire, pour tout titre  
de protection régionale disponible) : ARIPO (BW, GH,  
GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ,  
UG, ZM, ZW), eurasien (AM, AZ, BY, KG, KZ, RU, TJ,  
TM), européen (AL, AT, BE, BG, CH, CY, CZ, DE, DK,  
EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV,  
MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM,  
TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW,  
KM, ML, MR, NE, SN, TD, TG).

[Suite sur la page suivante]

(54) Title : METHOD FOR CREATING A CLIENT CONFIGURATION SIMULATOR

(54) Titre : PROCÉDÉ DE CRÉATION DE SIMULATEUR DE CONFIGURATION CLIENT

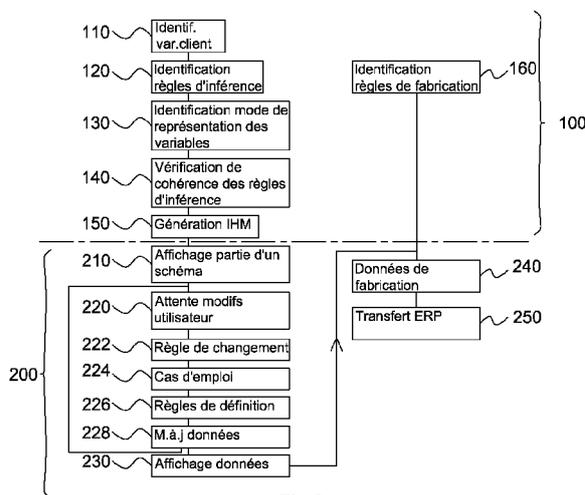


Fig.2

- 110 Identification of client var.
- 120 Identification of inference rules
- 130 Identification of mode of representation of variables
- 140 Checking consistency of inference rules
- 150 Generation of IHM
- 160 Identification of production rules
- 210 Display of part of a diagram
- 220 Waiting for user modifications
- 222 Change rule
- 224 Use case
- 226 Definition rules
- 228 Updating of data
- 230 Display of data
- 240 Production rules
- 250 ERP transfer

(57) Abstract : The invention concerns a method for creating an integrated custom software application for a computer comprising means for communicating with an Internet data exchange network, said application comprising an MVC human-machine interface, consisting of a data model, a visual interface, and finally an event logic controller, said method comprising graphic means for allowing a user to develop said software application without directly writing computer code, the event logic controller imposing relations between the data of the model and the elements of the display according to a set of settable so-called "business rules", which are also defined by the user without the need to directly write lines of computer code.

(57) Abrégé : L'invention vise un procédé de création d'une application logicielle intégrée sur mesure pour ordinateur comportant des moyens de communication à un réseau d'échange de données de type Internet, ladite application comportant une interface homme-machine de type MVC, constituée d'une part d'un modèle de données, d'autre part d'une interface visuelle, enfin d'un contrôleur de logique des événements, ledit procédé comportant des moyens graphiques pour permettre à un utilisateur de développer ladite application logicielle sans écrire directement de code informatique, le contrôleur logique d'événements imposant des relations entre les données du modèle et les éléments de la visualisation selon un ensemble paramétrables de règles dites "règles métier", également définies par l'utilisateur sans nécessiter l'écriture directe de lignes de code informatique.



WO 2014/090514 A1

**Publiée :**

— avec rapport de recherche internationale (Art. 21(3))

## PROCÉDÉ DE CRÉATION DE SIMULATEUR DE CONFIGURATION CLIENT

La présente invention relève du domaine des procédés de simulation logicielle.

Elle concerne plus particulièrement un procédé de génération d'application logicielle permettant de manipuler des données variables liées  
5 entre elles, de façon conforme à des règles de cohérence.

### Exposé de l'invention

L'invention vise en premier lieu un procédé de création d'une application logicielle intégrée sur mesure pour ordinateur mobile comportant des moyens  
10 de communication à un réseau d'échange de données de type Internet,

ladite application comportant une interface homme-machine de type MVC (Model View Controller, méthode de conception d'interface homme-machine de logiciel), constituée d'une part d'un modèle de données, d'autre part d'une interface visuelle, enfin d'un contrôleur de logique des évènements,  
15 ledit procédé comportant des moyens graphiques pour permettre à un utilisateur de développer ladite application logicielle sans écrire directement de code informatique.

Le contrôleur logique d'évènements impose des relations entre les données du modèle et les éléments de la visualisation selon un ensemble  
20 paramétrable de règles de cohérence et de règles d'inférence entre ces données, dites "règles métier" et définissant un moteur d'inférence appliqué aux données du modèle et aux éléments de visualisation.

Avantageusement, le procédé comporte un module de définition par l'utilisateur de ces règles métier sans nécessiter l'écriture directe par ledit  
25 utilisateur de lignes de code informatique. Les données sur lesquelles doivent s'appliquer les règles de cohérence sont désignées visuellement sur l'interface homme-machine par l'utilisateur, et des règles de lien mathématique entre diverses données, ou de seuils maximums ou minimums sont définis par l'utilisateur.

L'invention vise sous un autre aspect un procédé de création d'un simulateur logiciel d'un produit pouvant être configuré selon différentes variantes selon des données définissant les besoins et contraintes d'un client,

ledit procédé étant destiné à être mis en œuvre par un ordinateur  
5 comportant des moyens de communication à un réseau d'échange de données de type Internet,

ladite application comportant une interface homme-machine de type MVC, comportant :

- un modèle de données,
- 10 - une interface visuelle,
- un contrôleur de logique des évènements,

ledit procédé comportant des premiers moyens graphiques de définition de liens logiques entre les données, pour permettre à un utilisateur de développer ledit simulateur logiciel sans écrire directement de code  
15 informatique,

le contrôleur logique des évènements imposant des relations entre les données du modèle et les éléments de la visualisation selon un ensemble paramétrables de règles dites "règles métier", définies par des seconds moyens graphiques de définition de liens logiques entre les données.

20 On comprend que les produits cités ici peuvent comprendre plusieurs pièces et / ou mécanismes dont les dimensions ou caractéristiques respectives dépendent les unes des autres.

On comprend que l'invention est un procédé permettant de créer et  
25 adapter des applications intégrées sur mesure pour les supports PC, terminal nomade ou réseau web pour permettre à un utilisateur de concevoir, de développer et de déployer une application sur terminal informatique (tablette PC, smartphone, PC), ladite application permettant, en temps réel ou en mode déconnecté, à des utilisateurs clients professionnels (par exemple des  
30 revendeurs de menuiserie, des concepteurs d'armoires électriques..), des utilisateurs commerciaux, du marketing ou du Bureau d'Etudes et Production, de configurer, gérer, éditer sur écran, imprimer et générer des données structurées vers leurs bases de données (ERP, MRP, PDM) à partir

d'informations des produits et services configurables obtenues en temps réel à travers des choix d'options et de variantes.

5 Dans une mise en œuvre particulière, le procédé met en œuvre une arborescence d'affichage, dans laquelle l'utilisateur choisit d'afficher uniquement une partie d'un schéma de données sur laquelle il veut travailler et le procédé applique des liens et des règles de cohérence et d'inférence simples uniquement sur les éléments constituant cette partie.

10 Avantageusement, le procédé comporte une étape de génération d'un ensemble de données dite "données de fabrication", générées à partir des données d'affichage et d'un ensemble de règles de fabrication, préalablement défini.

15 Dans une mise en œuvre particulière, le module de définition de règles métier comporte des règles de cohérence applicables à ces règles métier, de manière à mettre en évidence les conflits éventuels entre certaines de ces règles lors de leur génération par l'utilisateur.

20 L'invention vise sous un autre aspect un procédé de fabrication d'un produit pouvant être configuré selon différentes variantes selon des données définissant les besoins et contraintes d'un client, comportant un procédé de création d'un simulateur logiciel d'un produit tel qu'exposé plus haut.

25 L'invention vise sous encore un autre aspect un produit / programme d'ordinateur adapté à mettre en œuvre un procédé tel qu'exposé plus haut.

### **Présentation des figures**

30 Les caractéristiques et avantages de l'invention seront mieux appréciés grâce à la description qui suit, description qui expose les caractéristiques de l'invention au travers d'un exemple non limitatif d'application.

La description s'appuie sur les figures annexées qui représentent :

Figure 1 : une vue schématique des éléments impliqués dans la mise en œuvre d'un procédé conforme à la présente description,

Figure 2 : un organigramme des différentes étapes du procédé.

## 5 Description détaillée d'un mode de réalisation de l'invention

Le procédé est destiné à être mis en œuvre de façon logicielle par un ensemble de systèmes informatiques.

Le procédé comporte en premier lieu une phase 100 de création, par une  
10 entreprise 32, offreur de produits ou de services, d'une application configurée selon un ensemble de règles de présentation d'information et de règles de calcul, ces règles étant spécifiques au métier de l'entreprise 32, et étant usuellement désignées sous le nom de "**règles métier**" d'une entreprise (voir figures 1 et 2). Cette phase 100 de création est également appelée dans la  
15 suite de la description "**configuration logicielle**", le procédé étant alors cité sous le terme de "**procédé de création de simulateur de configuration client**".

On comprend qu'il s'agit d'une modélisation algorithmique des règles  
métier de l'entreprise 32, et de leur présentation sous une forme  
20 ergonomiquement astucieuse et ludique à un utilisateur 22 devant utiliser des données représentatives de variables liées au métier de l'entreprise 32. Le but est de parvenir à un système de simulation utilisable par un utilisateur 22, non expert dans le métier de l'entreprise 32, et cependant conforme à toutes ses règles de cohérence.

25 A titre d'exemple purement explicatif et nullement limitatif, l'entreprise 32 peut être un fabricant de portails. Ces portails sont réalisables selon diverses options, par exemple en un certain nombre de matériaux, formes et textures différents, et selon diverses variantes, par exemple en largeur et hauteur variables. Dans un exemple plus détaillé, ils sont éventuellement divisibles en  
30 deux portes, dont la somme des largeurs correspond à une largeur d'emplacement d'implantation du portail. Différents matériaux peuvent être utilisés pour les portes, dans la mesure du respect de certaines contraintes

techniques. Les hauteurs des deux bords latéraux du portail peuvent varier, dans des limites de construction.

On comprend que le but est ici de proposer un simulateur de configuration de portail, permettant à un client potentiel 22 de l'entreprise 32 de  
5 configurer un portail qui corresponde à ses données spécifiques et à ses envies, sans jamais avoir à se préoccuper des règles métier qui régissent les relations des variables définissant le portail entre elles. Un tel configurateur doit permettre de faire varier de façon aussi continue que possible les paramètres disponibles à l'utilisateur 22. Dans le même temps, une variable de coût de  
10 réalisation peut être associée en temps réel à chaque configuration.

Un second but est de générer ensuite de façon correcte un cahier des charges de fabrication mis en forme selon les règles de présentation des données pertinentes pour le fabricant 32.

15

Cette première phase 100 du procédé accepte donc en entrées des règles métier diverses et un mode de représentation graphique des données, et fournit en sortie un code source correspondant à un configurateur logiciel réalisant les buts spécifiés plus hauts, c'est-à-dire par exemple un logiciel de  
20 définition d'une configuration de portail répondant aux besoins et contraintes particulières de clients 22.

Dans cette phase 100 de création de configurateur logiciel, le procédé implique un premier système informatique 10, ici dit serveur de développement.  
25 Ce serveur de développement 10 comporte un écran d'affichage 11, et héberge une ou plusieurs applications logicielles. Le serveur d'applications 10 est par exemple un ordinateur de type PC, connu en soi. Il dispose éventuellement de moyens classiques de communication au réseau, par voie filaire et/ ou radioélectrique. Il est adapté à héberger en mémoire et exécuter  
30 une application logicielle représentant la partie du procédé mise en œuvre par ledit serveur de développement 10.

Le procédé comporte en second lieu une phase 200 d'utilisation de l'application logicielle créée, cette utilisation étant réalisée par exemple mais non limitativement dans les locaux de l'entreprise 32. Cette utilisation s'apparente, comme on l'a dit, à l'exécution d'un procédé de simulation. Ce  
5 procédé de simulation permet par exemple de configurer un produit destiné à la fabrication selon les besoins spécifiques d'un client 22, et intégrant les règles métier de l'entreprise 32, c'est à dire, par exemple, des règles de fabrication et de dimensionnement.

10 Le procédé implique en premier lieu, pour cette seconde phase 200, un système informatique 15, ici dit serveur d'applications (voir figure 1). Ce serveur d'applications 15 comporte un écran d'affichage 11, et héberge une ou plusieurs applications logicielles. Le serveur d'applications 15 est par exemple un ordinateur de type PC, connu en soi. Il dispose de moyens classiques de  
15 communication au réseau, par voie filaire et/ ou radioélectrique. Il est adapté à héberger en mémoire et exécuter une application logicielle représentant la partie du procédé mise en œuvre par ledit serveur d'applications 15. Ce serveur d'applications 15 héberge notamment l'application logicielle (dite configurateur) générée lors de la phase de configuration logicielle.

20 La seconde phase 200 du procédé implique en outre au moins un second système informatique, dit ici terminal utilisateur 20. Dans le présent exemple, ce terminal utilisateur 20 est de type PC.

Le terminal utilisateur 20 comporte des moyens de calcul (par exemple microprocesseur), des dispositifs mémoire, et des moyens de communication  
25 16 au réseau par voie radioélectrique. Il est adapté à télécharger et exécuter une application logicielle reprenant la partie du procédé mise en œuvre par ledit terminal utilisateur 20.

Un tel terminal utilisateur 20 est de type connu en soi, et n'est donc pas détaillé plus avant ici.

30 Dans un premier mode de réalisation, le terminal utilisateur 20 accède via un réseau de type Internet à un serveur d'application 15, sur lequel est exécutée l'application correspondant aux règles métier de l'entreprise 32.

Dans un second mode de réalisation, le terminal utilisateur 20 et le serveur d'application 15 sont confondus en une seule machine, sur laquelle est exécutée l'application logicielle (le simulateur de configuration).

5 On comprend que cette seconde phase 200 du procédé peut impliquer plusieurs terminaux utilisateurs 20, ou de nombreux serveurs d'applications 15, dont les applications (simulateurs de configuration) sont éventuellement totalement différentes et relatives à des produits indépendants de l'entreprise 32.

10 Dans un mode particulier, le procédé met également en œuvre un serveur de gestion de données d'entreprises ERP 30, associé à un ensemble de terminaux 35. Dans cette mise en œuvre, les données modifiées par les actions sur les terminaux utilisateurs 20 sur le serveur d'application 15 sont transférées vers le serveur ERP 30, par exemple sous forme de données de  
15 fabrication.

### **Etapes du procédé**

La figure 2 illustre un organigramme des étapes d'un procédé conforme à un mode de mise en œuvre de l'invention.

20 Le procédé de création de configurateur est un modélisateur de structures hiérarchiques des données de l'entreprise 32.

Il utilise en entrées une identification et une définition des variables à considérer dans l'application client, un ensemble de règles de relations entre ces variables (inférences, équations entre ces variables...), des choix de  
25 modes de représentation de ces variables pour un utilisateur 22 non spécialiste. Ces entrées sont demandées progressivement à l'entreprise 32 lors de la création du simulateur de configuration, avec une capacité permanente à revenir sur des entrées précédentes, et à modifier celles-ci, de façon interactive.

30

On définit ci-dessous les principaux modules entrant dans la mise en œuvre du procédé de configuration.

## **Module de représentation et de manipulation des données**

### **Structure hiérarchique de données**

Les données utilisées dans le procédé de configuration sont stockées et manipulées sous la forme d'un ou plusieurs structures hiérarchiques de données. La structure hiérarchique de données est la structure de base du 5 moteur de configuration. Cette structure hiérarchique de données est celle qui est utilisée dans le moteur d'inférence mis en œuvre dans le procédé.

La structure hiérarchique de données utilise des modules d'accès aux données et de liens entre les données pour son affectation et les ensembles de 10 définitions de chaque valeur.

La structure de données hiérarchique permet de stocker et de manipuler toutes les données de l'entreprise 32 avec une même logique sans se soucier de sa persistance physique et de la manière dont cela est mis en forme à 15 l'écran.

Chaque structure hiérarchique de données possède un nom en relation avec son schéma de données, c'est-à-dire une liste de champs fortement typés, et un ensemble d'autres structures de données, éventuellement de type différent.

20 On rappelle que, dans des **champs fortement typés**, les variables ne sont pas dynamiques comme en Javascript (marque déposée), elles ont un type strict (entier, chaîne de caractères, date).

### **Schéma de règles**

25 Le procédé de configuration utilise en second lieu un schéma décrivant les règles qui régissent une structure hiérarchique de données.

Un schéma de règles est ici défini comme la description de la structure et de l'ensemble des règles qui définissent une structure hiérarchique de données. Un schéma au format XSD peut être déduit d'un schéma de règles et 30 peut ainsi être utilisé dans les échanges http sur un réseau de communications de données de type Internet.

Dans cette figure, qui illustre un exemple non limitatif d'une partie du contenu d'un schéma de règles, on comprend qu'une variable d'un Type

Complexe ici nommée « Toy » est composée d'occurrences d'une variable d'un Type Complexe « Board ». L'ensemble d'occurrence des variables « Board » forme un ensemble de données. Cet ensemble de données peut être représenté sous forme de tableau.

- 5 On peut maîtriser cet ensemble d'occurrences aux travers de contraintes, par exemple de valeur minimum ou maximum.

Des règles d'inférences associées à chaque variable de type simple permettent de rendre la structure de données active aux changements de valeur. En effet la structure hiérarchique de données peut être associée à un  
10 moteur d'inférence qui permet la propagation et la résolution de l'induction provoquée par les deux règles associées à chaque type simple.

Une **règle de changement** fonctionne comme un évènement lors du  
15 changement de valeur d'une variable d'une structure de données hiérarchique, cette règle est donc une règle explicite. Il est à noter qu'il n'y a pas ici enchaînement en cascade comme en informatique (la règle modifiant à son tour d'autres variables qui déclenchent à leur tour leur règle explicite), on respecte au contraire les pas du moteur d'inférence. Contrairement aux règles  
20 de définition (voir plus bas), une règle de changement ne s'exécute que si la valeur de la variable a été modifiée, elle ne tient pas compte des relations avec les autres variables, dans les pas d'inférence la règle de changement est calculé avant les règles de définitions.

25 Une **règle de définition** permet de définir la valeur d'une variable d'une structure de données hiérarchique, en fonction de son environnement. On rappelle ici qu'une règle de définition est à la base de l'inférence, c'est une règle qui définit la valeur de la variable en fonction des autres variables. Contrairement à une règle de changement, une règle de définition est implicite  
30 (c'est à dire que, par exemple, dans l'égalité  $v1 = v2 - v3$ , si  $v2$  change  $v1$  doit être recalculé). Elle doit déterminer son cas d'emploi pour pouvoir s'exécuter au moment opportun. Pendant un pas d'inférence toutes les règles sont calculées sur le même ensemble de valeur (équipotence entre les valeurs et le

résultat), puis toutes les valeurs sont mises à jour, ainsi de suite jusqu'à la stabilisation de l'ensemble.

### **Identifiant unique de référence**

5 Un identifiant unique de référence (URI=Unique Reference Identifier) permet d'obtenir de manière unique une structure de données hiérarchique ou une donnée particulière dans une structure complexe. Cet identifiant unique de référence Uri est l'adresse de la variable dans une structure hiérarchique.

Un ensemble de règles d'écriture sur ces identifiants sont définies.

10 Une méthode de recherche « Find » permet de retrouver un élément en fonction de son identifiant unique de référence, une syntaxe particulière permet de rechercher une structure hiérarchique de données en fonction de la valeur d'un de ses champs.

Les identifiants uniques de référence servent principalement pour les  
15 piles, l'association d'une entité graphique (TextBox) avec une variable du modèle ("binding") et l'accès aux données dans les algorithmes complexes.

L'identifiant unique de référence Uri sert à connaître l'adresse de la variable associée.

Les algorithmes complexes sont de type WebRules ou BusinessRules,  
20 ce sont des règles algorithmiques qui permettent de lier des structures entre elles, par exemple de transformer une structure de données de configuration en une structure pour l'impression, ou pour injection dans un logiciel de type ERP, etc.

Une extension est éventuellement faite à une structure hiérarchique de  
25 données pour y ajouter un ou plusieurs index de recherche dans le cas d'ensemble conséquent de données, et pour s'appuyer sur la méthode connue en soi « Link ».

### **Outil de stockage de structures hiérarchiques de données**

30 Une structure hiérarchique de données, telle que définie précédemment, est un ensemble de données qui ne peut persister qu'en mémoire. Pour pouvoir sauvegarder toutes les structure hiérarchique de données qui composent une configuration (une configuration est représentable comme un

ensemble de collections de structures de données), on utilise, dans le présent procédé de configuration, un outil générique pour le stockage de ces collections de structures de données en mémoire persistante.

5 Cet outil générique est une collection de structures de données persistantes. Une collection de structures hiérarchiques de données est associée à un schéma de règles unique qui décrit le comportement de tous les éléments (structures hiérarchiques de données) de la collection.

10 Une collection de structures de données est également associée à un ensemble de fournisseurs de données (fournisseurs de contenus mis à disposition pour la réalisation de configurations) qui permettent le stockage physique de la collection en fonction de différentes stratégies de stockage.

15 Une collection de structures de données doit-être stockée sur disque, en base, dans des services de stockage délocalisés ("cloud"), ou autre suivant la stratégie, et chaque stratégie à son fournisseur (provider). On utilise ici par exemple des fournisseurs de données classiques, en mémoire, en base de données, mais d'autres fournisseurs peuvent être envisagés, comme le stockage de données dans des listes de type connu en soi sous le nom de SharePoint –marque déposée-, dans des services de stockage délocalisés  
20 (Cloud) ou des centres de stockage de données externes (Datacenters).

### **Module de gestion graphique et d'exposition sur le Web ("service web")**

25 Le procédé comporte en outre un module de représentation des données manipulées dans le module précédent, selon un format compatible avec un affichage sur des ordinateurs reliés à un réseau de type Internet. Cette exposition se fait sous forme de services Web en utilisant l'architecture de Microsoft (IIS 7.0, WCF, .Net 3.5) (marques déposées).

30 Dans un premier mode de réalisation, non limitatif, on utilise un service web qui respecte l'architecture REST (de l'anglais Representational State Transfert), connue en soi.

L'architecture REST utilise des standards, en particulier :

- URI comme syntaxe universelle pour adresser les ressources,
- HTTP un protocole sans état (en anglais stateless) avec un nombre très limité d'opérations,
- Des hyperliens dans des documents (X)HTML et XML pour représenter à la fois le contenu des informations et la transition entre états de l'application,
- Les types MIME comme text/xml, text/html, image/jpeg, application/pdf, video/mpeg pour la représentation des ressources.

10 Les choix technologiques d'implémentation du service web sont un serveur IIS 7.0 (marque déposée) comme plate-forme d'hébergement du service (serveur d'application 15), et, dans un exemple nullement limitatif de mise en œuvre du procédé, une écriture du service en WCF (Windows Communication Foundation) en utilisant un lien d'objets pour faire  
15 communiquer le modèle de données et la visualisation utilisateur ("binding") spécifique qui expose (c'est à dire génère une interface visuelle) ce service en respectant l'architecture REST. Il est à noter que WCF est un package de Microsoft (marque déposée) qui enferme des protocoles de communication, ce package est utilisé dans le présent procédé pour Silverlight (marque déposée),  
20 mais pas pour html5.

Le serveur WEB mis en œuvre dans le présent procédé propose des services au format conforme aux contraintes REST (Representational State Transfer) en XML ou JSON, ou éventuellement un autre format connu de l'homme du métier.

25 De façon résumée, les données sont obtenues d'un service Web pour être liées (au sens du lien d'objets pour faire communiquer le modèle de données et la visualisation utilisateur –binding-) aux contrôles visuels.

Le procédé comporte également un service web en charge la  
30 persistance et la gestion des configurations en cours. Il est bâti autour d'une architecture de caches configurables, de manière à donner à ce service Web une grande fluidité d'utilisation.

Ce service web contient des objets temporaires qui permettent

l'exécution de différentes configurations, chacune est identifiée par un ticket dont la durée de vie est gérée par une stratégie fonction de conditions spécifiques à l'application ou au client considérée.

5 Le procédé utilise un ensemble de descriptions des collections de structures de données qui composent la configuration.

Le procédé utilise en outre des descripteurs de collections de structures hiérarchiques de données partageant un même schéma de règles (c'est à dire associées au même moteur d'inférence). Chaque collection de structures  
10 hiérarchiques de données possède son propre fournisseur de données clients. On rappelle qu'une collection doit-être stockée sur disque, en base, sur le cloud, ou autre suivant la stratégie, et chaque stratégie à son fournisseur (provider).

Le procédé travaille sur un ensemble de variables de base (destinées à  
15 être accessibles à un utilisateur du configurateur).

Une modification de la valeur d'une variable de base entraîne le démarrage du moteur d'inférence (le module de manipulation de données, c'est à dire le schéma de règles). Ce module permet de définir des règles dans les bases de données qui provoquent un événement (exemple un ajout, une  
20 modification). Cet événement est lié au cache de données (mise à jour) ou au moteur d'inférence (modification d'une règle de définition).

Par exemple, à titre de clarification, si la valeur d'un stock est modifiée, les règles associées à la quantité restante sont mises à jour pendant la  
25 configuration.

Une variable de base est associée à des contraintes définies dans le schéma et à des règles vérifiées par le moteur d'inférence. Certaines variables de base sont associées à un paramètre qui définit leur ensemble des valeurs possibles.

30 Le procédé conserve, dans le présent exemple non limitatif de mise en œuvre, au moins une liste chronologique décrivant les variables de base modifiées par l'utilisateur ou le moteur d'inférence. Chaque variable de base est décrite notamment par la localisation de la variable de base ainsi que sa

valeur, ou par un ensemble de paires clef / valeurs possibles pour une variable de base, cette liste définit l'ensemble des valeurs possible d'une variable de base en fonction de l'état d'une structure hiérarchique de données.

5 Le procédé de création de configurateur

Une configuration effectuée pour un client 22 (par exemple conception d'un portail de dimensions et géométrie spécifiques) est une suite d'opérations effectuées au sein d'un flux de donnée semblable à un flux de travaux (en anglais workflow).

10 Certaines opérations sont de type logique et d'autres de type graphique (formulaire). La demanderesse a constaté qu'en affinant la granularité, un grand nombre de formulaires présentent des fonctionnements similaires, seule change la logique permettant au client de gérer le schéma des règles de la structure hiérarchique de données.

15

En partant de ces observations, le procédé de création de configurateur a pour objectif de proposer un certain nombre de modèles d'opérateurs logiques et d'objets graphiques qui peuvent s'assembler au sein d'un flux paramétrable.

20 Le procédé de création de configurateur utilise les données et fonctionnalités du module de manipulation de données, et sur les technologies connues sous les noms Wpf, Wcf, Wf.

25 Le procédé de création de configurateur en ligne s'appuie sur une architecture de type connu sous le nom de Multi-Tier (architecture à plusieurs niveaux), ce qui signifie que certains objets graphiques logiques sont hébergés dans des Services web. Le procédé de création de configurateur, décrit ici à titre d'exemple, offre un moyen de générer un configurateur clients/Serveur d'une manière simple et accessible.

30 Les modèles proposés doivent être souples pour répondre à toutes les configurations. Pendant une configuration complexe, il peut avoir plusieurs structures en inférence. Etant donné qu'il n'y a pas de données stockées sur le terminal client 20, le procédé comporte des sessions de configuration sur le

serveur associé, et à chaque modification de données le procédé fait un appel vers la session associée (cookie) qui infère et renvoie le résultat au terminal client 20. Dans une autre variante de mise en œuvre du procédé, on appelle une règle de type WebRule qui injecte des données d'une session à une autre  
5 en respectant un ensemble de règles, ainsi aucun calcul et aucune donnée ne se trouve stockée chez le client 20 (selon une philosophie de "client léger").

Dans un mode de mise en œuvre décrit ici à titre d'exemple non limitatif, les objets graphiques sont développés en langage Wpf, le paramétrage des  
10 objets graphiques se fait par un simple push de composant et un lien des objets entre eux (binding) pour permettre la communication entre le modèle de données et la visualisation utilisateur) semi automatique à la structure hiérarchique de données au sein du Studio (marque déposée).

Dans une autre mise en œuvre, ces structures hiérarchiques sont des  
15 modèles de données (dynamiques). Pour les associer à un affichage, le procédé comprend un module de présentation de données. Ce module de présentation de données a pour fonction de présenter les données (couleur, langue du client, étiquette associée, type de contrôle), et de les modifier à l'aide d'un outil graphique dans le Studio (marque déposée).

20 A partir de ce module de présentation de données, on génère un affichage (frame) dans la technologie d'affichage utilisée (Xaml ou html5, ou autre à venir) en appliquant le lien des objets entre eux (binding).

Pour lier tous les modules de présentation de données (associés chacun à une structure hiérarchique de données), le procédé met en œuvre une étape  
25 qui permet de naviguer d'un premier module de présentation de données à un second module de présentation de données, en suivant le processus désiré par le client. Dans cette étape, on s'assure du passage des identifiants de session, du retour au précédent et de la communication entre les modules de présentation de données.

30 L'affichage suit le flux des modules de présentation de données en activant la présentation ("frame") (formulaire ou UserControl) associée.

Cette manière de paramétrer les formulaires permet de rendre

accessible la technologie SilverLight (marque déposée) ou html5 (ou autre dans des variantes de mise en œuvre du procédé) très complexe et non accessible au non informaticien. Des feuilles de style sont fournies pour l'habillage des configurations, permettant la personnalisation de la configuration en quelques clics.

Le procédé utilise plusieurs types d'éléments distincts :

- environnement graphique de base, muni d'un menu interaction et d'un espace de travail contenant des objets graphiques.
- objet graphique de base fourni dans une bibliothèque.
- opérateur logique (objet de calcul) ayant une fonctionnalité précise.
- descripteur de commande, objet permettant (ou provoquant) la transition du flux de données d'un objet graphique vers un autre objet graphique ou un opérateur logique. Une commande provoque au travers d'une règle le passage d'un module de présentation de données à un autre, ou l'appel d'une WebRule (calcul sur le serveur)
- structure hiérarchique de données, objet permettant le stockage des données fournies par les différents composants

### **Fonctionnement du procédé**

Le fonctionnement du procédé est illustré par l'organigramme de la figure 2.

Dans une première étape 110 de la phase 100 de création de configurateur logiciel, l'entreprise 32 procède à l'identification des variables permettant la définition des produits qu'elle veut faire configurer par ses clients 22. Dans une seconde étape 120, les règles d'inférence sont identifiées, puis dans une troisième étape 130, les modes de représentation des variables sont définis. Le système procède alors, dans une étape 140, à une vérification de cohérence des règles d'inférence définies précédemment. Il génère alors, dans une étape 150, une interface visuelle destinée aux utilisateurs 22 du configurateur. Dans une étape 160, des règles de fabrication sont définies par l'entreprise 32, sur la base des variables définies dans l'étape 110.

Dans une étape 210 de la phase 200 d'utilisation du configurateur, le système affiche, à destination de l'utilisateur 22, au moins une partie d'un schéma de données.

Puis l'utilisateur 22 vient modifier des données en fonction de ses besoins propres, et le moteur d'inférence propage les modifications aux autres variables liées par des règles d'inférence.

Plus précisément, lors du fonctionnement du moteur d'inférence, le moteur passe à travers plusieurs états qui s'enchainent de manière cyclique jusqu'à la stabilisation de l'ensemble de données ou à la détection de la récurrence de données cycliques.

Dans une première étape 220, le système est en attente de modifications de la part d'un utilisateur 22 du procédé.

Lors de la réception d'une telle modification, dans une étape 222, le système exécute les règles de changement pertinentes. Les règles de changement sont exécutées pour toutes les valeurs se trouvant dans la pile de modification, chaque règle de changement peut changer d'autres données qui sont à leur tour stockées dans la prochaine pile de modification.

Puis dans une étape 224, le système recherche les cas d'emploi des règles de définition. La recherche des cas d'emploi se fait en plusieurs états : dans un premier temps toutes les règles de définitions sont exécutées pour déterminer les données utilisées pour chaque règle, par la suite ne sont exécutées que les règles liées aux valeurs modifiées.

Il exécute alors dans une étape 226 les règles de définition pertinentes. Les règles de définitions sont exécutées en fonction de la pile de modification et des cas d'emploi. Chaque règle de modification met à jour la donnée concernée qui vient s'ajouter dans la pile de modification si celle-ci est réellement différente de la valeur précédente.

Puis le système met à jour, dans une étape 228, l'ensemble des données. Toutes les données modifiées sont stockées dans une pile de modification avant d'être mémorisées et comparées à un historique des données. Cette pile de modification sert pour éviter que l'ensemble des données soit modifié pendant l'exécution des différentes règles. Chaque règle est exécutée à partir du même ensemble de définition, ce qui évite le problème

d'exécution de code sur un ensemble de données en modification.

Lorsque ces données sont stabilisées, la nouvelle configuration est affichée dans une étape 230 sur l'interface utilisateur. La stabilisation de l'ensemble de données est obtenue quand il n'y a plus de données modifiées (variation inférieure à un seuil prédéterminé). Dans le cas de cycle (données ne convergent pas vers un ensemble de valeurs, mais revenant régulièrement sur un même jeu de valeurs) une autre stratégie est déployée en tenant compte de l'historique des modifications. Pour détecter les cycles, on recherche, lors de la mise à jour des données, si un ensemble identique a existé dans le passé. Dans ce cas, des règles de choix, préalablement entrées par l'entreprise 32, sont utilisées pour la détermination du meilleur ensemble de données parmi les ensembles revenant régulièrement lors de l'exécution du moteur d'inférence.

Dans une étape 240, le système génère les données de fabrication du produit défini par l'utilisateur 22 et correspondant à ses besoins et contraintes propres.

Enfin, dans une étape optionnelle 250, le système transfère des données de calcul à un serveur de type ERP 30 de l'entreprise 32.

Le principe de fonctionnement du procédé décrit ici se distingue de l'art antérieur dans lequel on regarde une configuration comme l'ouverture d'une fiche au travers de code se trouvant derrière des boutons.

On imagine qu'une configuration est un flux applicatif dont la donnée est transportée par des structures hiérarchiques de données et le changement d'état (objet graphique ou opérateur logique) provoqué par des commandes. Dans le procédé décrit ici, les flux peuvent se multiplier et fusionner dans certain opérateur logique, ce qui donne la possibilité d'avoir plusieurs états en même temps, c'est-à-dire plusieurs objets graphiques à la fois.

On observe sur ce graphique le flux de déroulement de la configuration, l'objet graphique passe la main à l'opérateur logique au travers d'une commande, éventuellement soumise à des droits utilisateur.

Le procédé utilise un environnement graphique de base qui héberge tout les autres composants, il est formé de trois zones :

- un bandeau d'affichage type Office 2007 (marque déposée) interactif et contextuel,
- un objet de présentation sous forme d'onglet, et
- un espace de travail qui héberge les différents objets graphiques.

5

L'environnement graphique de base est optimisé pour fonctionner aussi bien sur un poste de travail que dans un explorateur internet.

L'environnement graphique de base n'est pas obligatoire dans une configuration, mais il est souvent l'objet de démarrage, et il contient les  
10 commandes de bases qui démarrent le flux applicatif.

L'environnement graphique de base héberge des objets graphiques de base. Chaque tel objet graphique contient un espace de travail, et une partie de ruban (un ruban est un élément d'interface utilisateur, introduit par Microsoft  
15 –marque déposée-, qui remplace ou complète les boutons des menus connus précédemment et permet de regrouper des commandes associées pour les retrouver plus facilement) qui complètera le ruban de l'environnement graphique de base.

Il existe une bibliothèque d'objets graphiques spécialisés (saisie de  
20 données, recherche, ...), un outil est proposé pour la conception d'un objet graphique, celui-ci étant associé à une structure hiérarchique de données, qui lui sert de contexte et une collection de structures hiérarchiques de données (associée à un schéma de règles unique) pour l'accès aux données, un mécanisme simple de push de composant est proposé au concepteur avec le  
25 minimum de variable de paramétrage.

Le ruban associé à l'objet graphique est un assemblage de groupe et de descripteurs de commandes qui permet de respecter le mécanisme MVC (Modèle Vue contrôleur, en anglais "Model View Controler") et ce sont les commandes associées à chaque bouton qui permettent de naviguer d'objet  
30 graphique en objet graphique. On rappelle que **MVC** est une méthode de conception d'interface homme-machine de logiciel. Dans cette méthode, l'interface homme-machine comporte d'une part un modèle de données, d'autre part une interface visuelle, enfin un contrôleur de logique des évènements.

Il est clair que, dans des variantes de mise en œuvre, le procédé peut également utiliser d'autres méthodes que MVC pour naviguer entre les objets graphiques.

Les objets graphiques sont spécialisés pour résoudre des problèmes  
5 spécifiques, mais restent paramétrables au travers des règles associées à leur structure hiérarchique de données, et aménageables au travers de StackPanel de Microsoft –marques déposées- (commande permettant de réorganiser des éléments secondaires dans une hiérarchie d'éléments, sur une seule ligne pouvant être orientée horizontalement ou verticalement) qui sont enrichies de  
10 composants par des actions de glisser / déposer d'éléments disponibles de la structure hiérarchique de données, sous forme de contrôle.

Chaque **opérateur logique** (objet de calcul) est un composant de liaison entre deux objets graphiques ou opérateurs logiques. On peut citer, par  
15 exemple mais non limitativement, des opérateurs logiques du genre envoi d'un mail, écriture dans un fichier, recherche d'information dans un Web Service.

Une bibliothèque d'opérateurs logiques est fournie dans le procédé. Cette bibliothèque d'opérateurs logiques évolue dans le temps.

La donnée traitée dans un opérateur logique est une structure  
20 hiérarchique de données, et la poursuite du flux se fait au travers de commandes qui sont exécutés par des déclencheurs ou des conditions programmables.

Dans l'exemple simple de création d'une nomenclature dans un logiciel de planification des ressources de l'entreprise (ERP), il existe un opérateur  
25 logique spécialisé pour cette opération, la structure hiérarchique de données en entrée est un extrait de la configuration, l'opérateur logique possède sa propre structure hiérarchique de données de nomenclature.

Une injection d'une structure hiérarchique de données a lieu pour la création de la nomenclature dans l'ERP. **L'injection** est le terme que l'on  
30 emploie pour transférer des données d'une structure à une autre. A titre de clarification, on peut avoir une structure complexe pour la configuration d'un objet, et venir injecter le résultat (données significatives) dans une autre structure (une occurrence de devis par exemple).

Une fois l'opération terminée, une commande est déclenchée pour la poursuite du flux.

Le procédé utilise un ensemble de descripteurs de commande. Chaque commande est associée à une image, un état, une description et une implémentation. Les descripteurs de commande sont regroupés dans un gestionnaire de commande indépendant. Au sens du présent procédé, un descripteur de commande utilise en entrée le contexte de l'objet graphique courant et fournit un contexte à l'objet graphique suivant. Comme les objets graphiques peuvent s'empiler dans certaines configurations, certaines commandes provoquent la simple fermeture d'un objet graphique.

Pour clarifier la notion d'empilement des objets graphiques, on note que dans le procédé, on a un système de navigation dans lequel l'utilisateur peut avancer dans le processus, mais aussi revenir en arrière, ou au départ.

On a donc un fil d'Ariane (ou empilement) composé de module de présentation de données, et certaine commande permettent de manipuler ce fil, (suivant, retour, fin, revenir au début, ...)

Une commande peut-être l'ouverture d'une fiche client, la commande doit extraire le code du client du contexte courant pour positionner l'objet graphique édition du client sur la bonne personne.

Le transport des données dans le flux se fait au travers de structures (ou conteneurs) hiérarchiques de données, les données peuvent transiter soit par la mémoire, soit au travers de collections persistantes de structures hiérarchiques de données.

### **Avantages**

Le procédé décrit plus haut présente plusieurs avantages :

Il permet de séparer les raisonnements sur les relations entre les données de l'affichage et du stockage physique. Tout cela en restant compatible avec les standards mondiaux XSD (XML schéma), ce qui permet une grande compatibilité avec les systèmes existants ou futurs.

Le procédé de configuration logicielle permet de définir les schémas de

données d'une manière beaucoup plus aisée que les systèmes de gestion de bases de données (SGBD) actuels, et de valider de manière implicite les données et les relations d'inférence des données entre elles.

Le concepteur 32 manipule ainsi plus facilement un arbre logique ; c'est  
5 plus simple et plus intuitif que de faire des rapprochements de vue au moyen de jointure.

Dans un exemple de mise en œuvre réel du procédé, la demanderesse a constaté que certaines configurations comportent huit mille types dans un schéma, ces huit mille types étant liés de manière implicite (arborescence).

10 Dans le cas d'une utilisation d'un système de gestion de base de données pour travailler sur cet ensemble, cela se traduit par huit mille tables et beaucoup plus de jointures qu'il faut appréhender en bloc.

Au contraire, dans l'arborescence utilisée dans le procédé, on affiche uniquement la partie sur laquelle l'on veut travailler et on applique des liens et  
15 des règles simples (on divise et on structure) sur les éléments constituant cette partie, ce que ne propose aucun système de gestion de base de données, lesquels travaillent à plat avec des jointures.

La traduction d'un schéma peu complexe en UML, sous forme système de gestion de base de données, se traduirait, pour l'utilisateur, en un ensemble  
20 considérable de blocs et de traits se croisant, probablement incompréhensible car présenté dans son ensemble.

Une première partie innovante du procédé, visible par l'utilisateur expert de l'entreprise 32 non informaticien, concerne la méthode logique de  
25 modélisation de règles métiers ordonnées en structure hiérarchique dans différents schémas. Les données intelligentes qui représentent les règles métier sont décrites dans le schéma et s'appuient sur des interfaces de visualisation et de liaisons.

Une seconde partie innovante, concerne le moteur d'inférence qui régit  
30 les schémas de configuration et les milliers de combinaison de règles associées, en temps réel à chaque changement d'état des données produits introduite par l'utilisateur final.

Une troisième partie innovante concerne la génération automatique de code informatique permettant ainsi la génération d'applications de configurations métiers à travers un studio ludique pour des experts métiers non-informaticiens.

- 5 Une quatrième partie innovante concerne la conception dynamique de formulaires métiers pilotés par les schémas. Ces formulaires sont générés en fonction de la plate-forme cible des postes clients (Silverlight, Windows8, HTML5 – marques déposées).

- 10 Ce moteur et les règles associées aux schémas sont résidents sur serveurs WEB dans le CLOUD et sont accessibles aux PC portables ou tablettes utilisateurs en mode multi-plateforme Windows, Androïd et IPAD (marques déposées).

## REVENDICATIONS

1. Procédé de création d'un générateur d'un ensemble de données dites "données de fabrication" d'un produit pouvant être configuré selon différentes variantes selon des paramètres définissant les besoins et contraintes d'un client (22), lesdits paramètres pouvant varier de façon continue,

5           ledit procédé étant destiné à être mis en œuvre par un ordinateur (20) comportant des moyens de communication à un réseau d'échange de données de type Internet,

          ladite application comportant une interface homme-machine de type MVC, comportant

- 10           - un modèle de données,  
          - une interface visuelle,  
          - un contrôleur de logique des évènements,

caractérisé en ce que ledit procédé comporte des premiers moyens graphiques de définition de liens logiques entre les données, pour permettre à  
15 un utilisateur (32) de développer ledit générateur de données de fabrication,

          le contrôleur logique des évènements imposant des relations entre les données du modèle et les éléments de la visualisation selon un ensemble paramétrables de règles dites "règles métier", définies par des seconds moyens graphiques de définition de liens logiques entre les données.

20

          2. Procédé selon la revendication 1, caractérisé en ce que le procédé comporte un module de définition par l'utilisateur (32) de ces règles métier, les données sur lesquelles doivent s'appliquer les règles de cohérence étant désignées visuellement sur l'interface homme-machine par l'utilisateur (32), et  
25 des règles de lien mathématique entre diverses données, ou de seuils maximums ou minimums étant définis par l'utilisateur (32).

          3. Procédé selon l'une quelconque des revendications 1 à 2, caractérisé en ce que le procédé met en œuvre une arborescence d'affichage, dans  
30 laquelle l'utilisateur (32) choisit d'afficher uniquement une partie d'un schéma de données sur laquelle il veut travailler et le procédé applique des liens et des

règles de cohérence et d'inférence simples sur les éléments constituant cette partie.

4. Procédé selon l'une quelconque des revendications 2 à 3, caractérisé en ce que le module de définition de règles métier comporte des règles de cohérence applicables à ces règles métier, de manière à mettre en évidence les conflits éventuels entre certaines de ces règles lors de leur génération par l'utilisateur (32).

5. Procédé de fabrication d'un produit pouvant être configuré selon différentes variantes selon des paramètres définissant les besoins et contraintes d'un client, lesdits paramètres pouvant varier de façon continue, caractérisé en ce qu'il comporte un procédé de création d'un générateur de données de fabrication du produit, selon l'une quelconque des revendications 1 à 4.

6. Procédé selon la revendication 5, caractérisé en ce qu'il comporte des phases et étapes suivantes :

phase 100 de création d'un générateur de données de fabrication d'un produit,

- 110 : identification de variables permettant la définition de produits qu'on veut faire configurer par des clients 22.

- 120 : identification des règles d'inférence

- 140 : vérification de cohérence des règles d'inférence définies précédemment.

- 160 : définition de règles de fabrication, sur la base des variables définies dans l'étape 110.

phase 200 d'utilisation du configurateur,

- 210 : affichage à destination d'un utilisateur 22, au moins une partie d'un schéma de données,

- 220 : modification par l'utilisateur des données en fonction de ses besoins propres, et

- propagation par le moteur d'inférence des modifications aux autres

variables liées par des règles d'inférence.

5 7. Procédé selon la revendication 6, caractérisé en ce que lors de l'étape de propagation d'une modification par le moteur d'inférence, le moteur passe à travers plusieurs états qui s'enchainent de manière cyclique jusqu'à la stabilisation de l'ensemble de données ou à la détection de la récurrence de données cycliques.

1/1

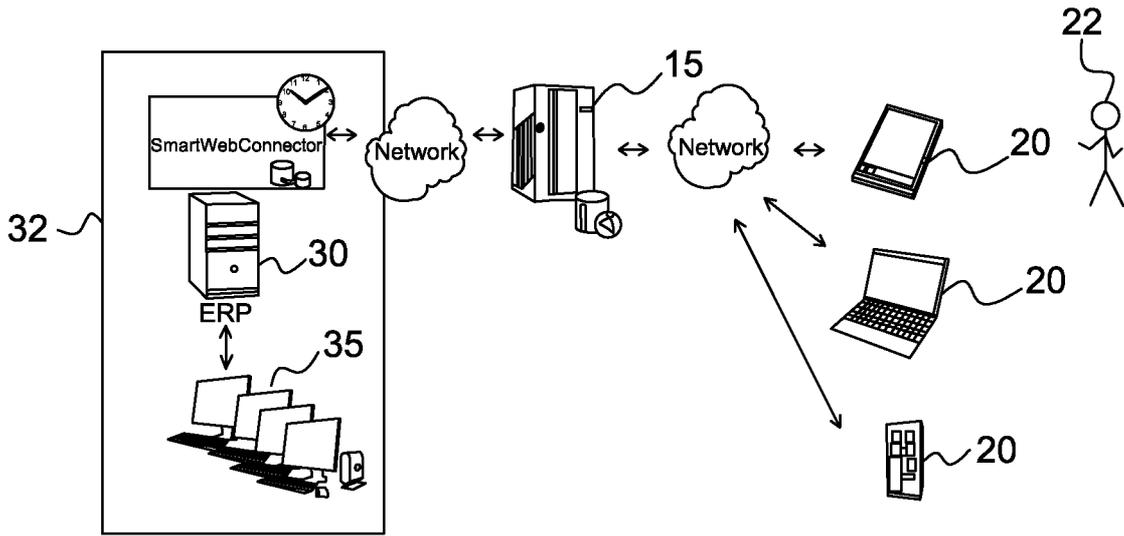


Fig.1

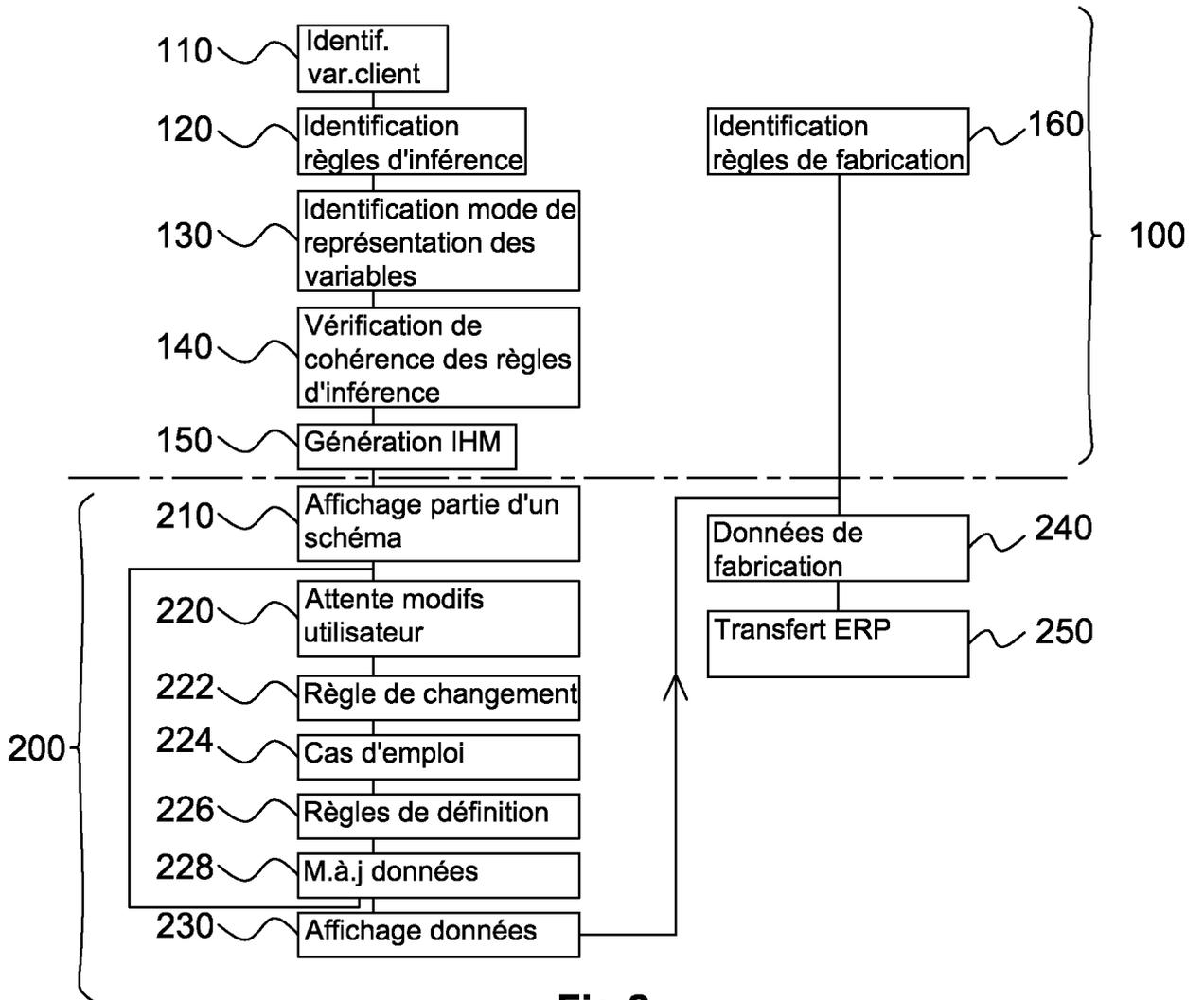


Fig.2

# INTERNATIONAL SEARCH REPORT

International application No <b>PCT/EP2013/074093</b>
--

<b>A. CLASSIFICATION OF SUBJECT MATTER</b> INV. G06F9/44                      G06F17/50 ADD.				
According to International Patent Classification (IPC) or to both national classification and IPC				
<b>B. FIELDS SEARCHED</b>				
Minimum documentation searched (classification system followed by classification symbols) G06F				
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched				
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) EPO-Internal, WPI Data, IBM-TDB, INSPEC, COMPENDEX				
<b>C. DOCUMENTS CONSIDERED TO BE RELEVANT</b>				
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.		
X	US 5 844 554 A (GELLER SCOTT D [US] ET AL) 1 December 1998 (1998-12-01) abstract; figures 2-4,6,7 column 1, line 5 - column 1, line 10 column 3, line 20 - column 5, line 35 column 7, line 40 - column 8, line 17 column 8, line 51 - column 9, line 5 column 10, line 7 - column 11, line 27 column 12, line 21 - column 13, line 65 column 15, line 20 - column 27, line 12 ----- -/--	1-7		
<input checked="" type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex.				
* Special categories of cited documents : <table style="width: 100%; border: none;"> <tr> <td style="width: 50%; border: none; vertical-align: top;">                     "A" document defining the general state of the art which is not considered to be of particular relevance                      "E" earlier application or patent but published on or after the international filing date                      "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)                      "O" document referring to an oral disclosure, use, exhibition or other means                      "P" document published prior to the international filing date but later than the priority date claimed                 </td> <td style="width: 50%; border: none; vertical-align: top;">                     "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention                      "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone                      "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art                      "&amp;" document member of the same patent family                 </td> </tr> </table>			"A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family
"A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family			
Date of the actual completion of the international search	Date of mailing of the international search report			
18 February 2014	26/02/2014			
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Authorized officer  Lelait, Sylvain			

INTERNATIONAL SEARCH REPORT

International application No  
PCT/EP2013/074093

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>US 2007/073763 A1 (BETTS CHRISTOPHER [AU] ET AL) 29 March 2007 (2007-03-29) abstract; figures 1,2 page 1, paragraph 2 - page 1, paragraph 4 page 1, paragraph 8 - page 1, paragraph 9 page 1, paragraph 13 - page 2, paragraph 15 page 2, paragraph 21 page 2, paragraph 23 - page 3, paragraph 32; claims 1-3,7,9,10,12,14-16,20,23,25,27-29,33,35,36,38</p>	1-7
A	<p>----- P. T. HELO ET AL: "Citarasa driven vehicle configurator", 2007 IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL ENGINEERING AND ENGINEERING MANAGEMENT, 1 December 2007 (2007-12-01), pages 1302-1306, XP055089493, DOI: 10.1109/IEEM.2007.4419403 ISBN: 978-1-42-441529-8 abstract page 1302, left-hand column, paragraph 2 - page 1302, right-hand column, paragraph 3 page 1303, left-hand column, paragraph 1 page 1303, left-hand column, last paragraph - page 1305, left-hand column, paragraph 1; figures 3-8</p>	1-7
A	<p>----- WO 01/29687 A1 (OFFICEFREE COM [US]) 26 April 2001 (2001-04-26) abstract page 1, line 5 - page 1, line 6 page 2, line 28 - page 3, line 19 page 5, line 1 - page 5, line 21 page 6, line 9 - page 6, line 20 page 7, line 19 - page 9, line 11</p>	1-7
A	<p>----- WO 2011/104367 A2 (SITA INFORMATION NETWORKING COMPUTING IRELAND LTD [IE]; FINDLAY DENISE) 1 September 2011 (2011-09-01) abstract page 2, line 17 - page 3, line 10 page 4, line 10 - page 4, line 24 page 5, line 32 - page 13, line 23 page 15, line 10 - page 15, line 30; claims 1-3,5,7,10,11; figure 1 ----- -/--</p>	1-7

# INTERNATIONAL SEARCH REPORT

International application No  
PCT/EP2013/074093

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2010/262902 A1 (BURNS STEVEN P [US]) 14 October 2010 (2010-10-14) abstract; figures 2-4,6,7 page 1, line 5 page 1, paragraph 13 - page 1, paragraph 17 page 2, paragraph 25 - page 3, paragraph 46 page 3, paragraph 51 - page 5, paragraph 79  -----	1-7

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No PCT/EP2013/074093
---

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5844554 A	01-12-1998	US 5844554 A US 6300948 B1	01-12-1998 09-10-2001
US 2007073763 A1	29-03-2007	US 2007073763 A1 WO 2007038352 A2	29-03-2007 05-04-2007
WO 0129687 A1	26-04-2001	US 2002078435 A1 WO 0129687 A1	20-06-2002 26-04-2001
WO 2011104367 A2	01-09-2011	AU 2011219701 A1 CA 2790985 A1 CN 103109266 A EP 2539811 A2 GB 2481191 A WO 2011104367 A2	13-09-2012 01-09-2011 15-05-2013 02-01-2013 21-12-2011 01-09-2011
US 2010262902 A1	14-10-2010	NONE	

# RAPPORT DE RECHERCHE INTERNATIONALE

Demande internationale n°  
PCT/EP2013/074093

<b>A. CLASSEMENT DE L'OBJET DE LA DEMANDE</b> INV. G06F9/44 G06F17/50 ADD.		
Selon la classification internationale des brevets (CIB) ou à la fois selon la classification nationale et la CIB		
<b>B. DOMAINES SUR LESQUELS LA RECHERCHE A PORTE</b>		
Documentation minimale consultée (système de classification suivi des symboles de classement) G06F		
Documentation consultée autre que la documentation minimale dans la mesure où ces documents relèvent des domaines sur lesquels a porté la recherche		
Base de données électronique consultée au cours de la recherche internationale (nom de la base de données, et si cela est réalisable, termes de recherche utilisés) EPO-Internal, WPI Data, IBM-TDB, INSPEC, COMPENDEX		
<b>C. DOCUMENTS CONSIDERES COMME PERTINENTS</b>		
Catégorie*	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
X	US 5 844 554 A (GELLER SCOTT D [US] ET AL) 1 décembre 1998 (1998-12-01) abrégé; figures 2-4,6,7 colonne 1, ligne 5 - colonne 1, ligne 10 colonne 3, ligne 20 - colonne 5, ligne 35 colonne 7, ligne 40 - colonne 8, ligne 17 colonne 8, ligne 51 - colonne 9, ligne 5 colonne 10, ligne 7 - colonne 11, ligne 27 colonne 12, ligne 21 - colonne 13, ligne 65 colonne 15, ligne 20 - colonne 27, ligne 12  ----- -/--	1-7
<input checked="" type="checkbox"/>	Voir la suite du cadre C pour la fin de la liste des documents	<input checked="" type="checkbox"/> Les documents de familles de brevets sont indiqués en annexe
* Catégories spéciales de documents cités:		
"A" document définissant l'état général de la technique, non considéré comme particulièrement pertinent "E" document antérieur, mais publié à la date de dépôt international ou après cette date "L" document pouvant jeter un doute sur une revendication de priorité ou cité pour déterminer la date de publication d'une autre citation ou pour une raison spéciale (telle qu'indiquée) "O" document se référant à une divulgation orale, à un usage, à une exposition ou tous autres moyens "P" document publié avant la date de dépôt international, mais postérieurement à la date de priorité revendiquée "T" document ultérieur publié après la date de dépôt international ou la date de priorité et n'appartenant pas à l'état de la technique pertinent, mais cité pour comprendre le principe ou la théorie constituant la base de l'invention "X" document particulièrement pertinent; l'invention revendiquée ne peut être considérée comme nouvelle ou comme impliquant une activité inventive par rapport au document considéré isolément "Y" document particulièrement pertinent; l'invention revendiquée ne peut être considérée comme impliquant une activité inventive lorsque le document est associé à un ou plusieurs autres documents de même nature, cette combinaison étant évidente pour une personne du métier "&" document qui fait partie de la même famille de brevets		
Date à laquelle la recherche internationale a été effectivement achevée  18 février 2014		Date d'expédition du présent rapport de recherche internationale  26/02/2014
Nom et adresse postale de l'administration chargée de la recherche internationale Office Européen des Brevets, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016		Fonctionnaire autorisé  Lelait, Sylvain

RAPPORT DE RECHERCHE INTERNATIONALE

Demande internationale n°  
PCT/EP2013/074093

C(suite). DOCUMENTS CONSIDERES COMME PERTINENTS		
Catégorie*	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
A	<p>US 2007/073763 A1 (BETTS CHRISTOPHER [AU] ET AL) 29 mars 2007 (2007-03-29) abrégé; figures 1,2 page 1, alinéa 2 - page 1, alinéa 4 page 1, alinéa 8 - page 1, alinéa 9 page 1, alinéa 13 - page 2, alinéa 15 page 2, alinéa 21 page 2, alinéa 23 - page 3, alinéa 32; revendications 1-3,7,9,10,12,14-16,20,23,25,27-29,33,35,36,38</p>	1-7
A	<p>----- P. T. HELO ET AL: "Citarasa driven vehicle configurator", 2007 IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL ENGINEERING AND ENGINEERING MANAGEMENT, 1 décembre 2007 (2007-12-01), pages 1302-1306, XP055089493, DOI: 10.1109/IEEM.2007.4419403 ISBN: 978-1-42-441529-8 abrégé page 1302, colonne de gauche, alinéa 2 - page 1302, colonne de droite, alinéa 3 page 1303, colonne de gauche, alinéa 1 page 1303, colonne de gauche, dernier alinéa - page 1305, colonne de gauche, alinéa 1; figures 3-8</p>	1-7
A	<p>----- W0 01/29687 A1 (OFFICEFREE COM [US]) 26 avril 2001 (2001-04-26) abrégé page 1, ligne 5 - page 1, ligne 6 page 2, ligne 28 - page 3, ligne 19 page 5, ligne 1 - page 5, ligne 21 page 6, ligne 9 - page 6, ligne 20 page 7, ligne 19 - page 9, ligne 11</p>	1-7
A	<p>----- W0 2011/104367 A2 (SITA INFORMATION NETWORKING COMPUTING IRELAND LTD [IE]; FINDLAY DENISE) 1 septembre 2011 (2011-09-01) abrégé page 2, ligne 17 - page 3, ligne 10 page 4, ligne 10 - page 4, ligne 24 page 5, ligne 32 - page 13, ligne 23 page 15, ligne 10 - page 15, ligne 30; revendications 1-3,5,7,10,11; figure 1</p>	1-7
A	<p>----- US 2010/262902 A1 (BURNS STEVEN P [US]) 14 octobre 2010 (2010-10-14) abrégé; figures 2-4,6,7 page 1, ligne 5 page 1, alinéa 13 - page 1, alinéa 17 page 2, alinéa 25 - page 3, alinéa 46 page 3, alinéa 51 - page 5, alinéa 79</p> <p>-----</p>	1-7

# RAPPORT DE RECHERCHE INTERNATIONALE

Renseignements relatifs aux membres de familles de brevets

Demande internationale n°

PCT/EP2013/074093

Document brevet cité au rapport de recherche		Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
US 5844554	A	01-12-1998	US 5844554 A US 6300948 B1	01-12-1998 09-10-2001
US 2007073763	A1	29-03-2007	US 2007073763 A1 WO 2007038352 A2	29-03-2007 05-04-2007
WO 0129687	A1	26-04-2001	US 2002078435 A1 WO 0129687 A1	20-06-2002 26-04-2001
WO 2011104367	A2	01-09-2011	AU 2011219701 A1 CA 2790985 A1 CN 103109266 A EP 2539811 A2 GB 2481191 A WO 2011104367 A2	13-09-2012 01-09-2011 15-05-2013 02-01-2013 21-12-2011 01-09-2011
US 2010262902	A1	14-10-2010	AUCUN	