

(19) RÉPUBLIQUE FRANÇAISE  
INSTITUT NATIONAL  
DE LA PROPRIÉTÉ INDUSTRIELLE  
PARIS

(11) Nº de publication :  
(à n'utiliser que pour les  
commandes de reproduction)

3 004 556

(21) Nº d'enregistrement national :

13 53213

(51) Int Cl<sup>8</sup> : G 06 F 17/24 (2013.01), G 06 Q 10/00

(12)

## DEMANDE DE BREVET D'INVENTION

A1

(22) Date de dépôt : 10.04.13.

(30) Priorité :

(43) Date de mise à la disposition du public de la demande : 17.10.14 Bulletin 14/42.

(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 :

Demande(s) d'extension :

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

(72) Inventeur(s) : FICET ALINE et DOUCERAIN CYRIL.

(73) Titulaire(s) : BULL Société par actions simplifiée.

(74) Mandataire(s) : CABINET HIRSCH & PARTNERS.

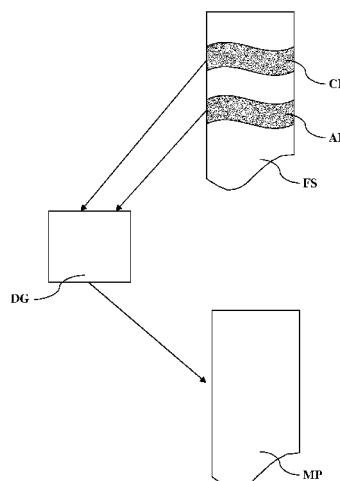
(54) GENERATION AUTOMATIQUE DE DOCUMENTATION.

(57) La présente invention concerne un dispositif de génération de documentation (DG) apte à générer de la documentation associée à un programme logiciel présentant un moyen d'analyse permettant de saisir des commandes comportant des arguments, ledit dispositif de génération de documentation comprenant un moyen pour

- Récupérer des premières informations (AI) connexes à un ensemble d'arguments possibles;

- Récupérer des secondes informations (CI) connexes à une commande, lesdites secondes informations comprenant des données de référence indiquant une liste d'arguments pour ladite commande;

- Générer de la documentation associée à ladite commande en récupérant un sous-ensemble desdites premières informations, déterminé selon ladite liste d'arguments, et fusionner au moins une partie desdites secondes informations avec ledit sous-ensemble desdites premières informations.



FR 3 004 556 - A1



## **GENERATION AUTOMATIQUE DE DOCUMENTATION**

### **DOMAINE DE L'INVENTION**

5

La présente invention concerne la génération de la documentation associée à un programme informatique doté de moyens permettant l'analyse d'une commande présentant des arguments.

### **10 CONTEXTE DE L'INVENTION**

Les produits logiciels comprennent un ou plusieurs programmes logiciels accompagnés de la documentation associée destinée à guider les utilisateurs lors de l'utilisation des programmes et/ou à aider à interfaire ces programmes avec d'autres programmes.

La rédaction de la documentation représente une tâche fastidieuse pouvant être réalisée par des personnes dévouées au sein d'une grande entreprise de produits logiciels dont les ventes et/ou les recettes sont importantes. Cependant, dans les entreprises de moindre envergure, ou pour les programmes logiciels dont les ventes ou les recettes sont limitées, les développeurs de logiciels sont souvent également chargés de la rédaction de la documentation associée. C'est par exemple le cas pour les programmes de système tels que ceux exécutés à partir d'un interpréteur de ligne de commande (CLI).

En outre, les programmes logiciels d'un même produit ou d'une même famille de produits peuvent être développés par différents développeurs, de sorte que la documentation associée peut être rédigée à différents moments par différentes personnes. Par ailleurs, de nouvelles versions d'un même programme logiciel peuvent être déployées et avoir une incidence sur la façon d'utiliser le programme, de sorte que la documentation doit également être mise à jour.

Par conséquent, de nombreuses différences sont constatées dans la documentation des programmes logiciels d'un même produit, par exemple pour les commandes disponibles sur un même interpréteur de ligne de commande. Les différences peuvent être d'ordre formel (par exemple lorsque les informations ne sont pas présentées de la même façon), mais elles peuvent également avoir une incidence sur des aspects plus importants, tel que le niveau de détails, etc., de sorte que l'utilisateur peut en être dérouté.

En outre, s'agissant d'une tâche fastidieuse pour le développeur ou la personne chargée de la rédaction de la documentation, cette dernière n'est pas toujours

complète, et il est possible que certains fragments d'information importants manquent à l'appel.

## 5 RÉSUMÉ DE L'INVENTION

La présente invention cherche à pallier, au moins partiellement, les inconvénients susmentionnés.

Cet objectif est atteint au moyen d'un procédé destiné à générer la documentation associée à un programme logiciel doté d'un moyen d'analyse permettant de saisir des commandes comportant des arguments, le procédé comprenant les étapes ci-dessous consistant à :

- Fournir des premières informations connexes à un ensemble d'arguments possibles ;
- 15 - Fournir des secondes informations connexes à une commande, lesdites secondes informations comprenant des données de référence indiquant une liste d'arguments pour ladite commande ;
  - Générer de la documentation associée à ladite commande en récupérant un sous-ensemble desdites premières informations déterminé selon ladite liste d'arguments, et fusionner au moins une partie desdites secondes informations avec ledit sous-ensemble desdites premières informations.

Les modes de réalisation préférés comprennent une ou plusieurs des caractéristiques suivantes, lesquelles peuvent être utilisées individuellement ou 25 communément, sous une forme partiellement ou totalement combinée :

- Le programme logiciel est un interpréteur de ligne de commande (CLI) ;
- Lesdites commandes comprennent au moins deux types d'arguments ;
- Lesdites premières informations et lesdites secondes informations associent des éléments et des valeurs, et lesdits éléments sont organisés de façon hiérarchique ;
- 30 - Un niveau de la hiérarchie desdites premières informations (AI) correspond auxdits types d'arguments ;
- Ladite documentation est une page de manuel.

35 Un autre aspect de la présente invention concerne un dispositif de génération de documentation apte à générer de la documentation associée à un programme logiciel présentant un moyen d'analyse permettant de saisir des commandes

comportant des arguments, ledit dispositif de génération de documentation comprenant un moyen pour :

- Récupérer des premières informations connexes à un ensemble d'arguments possibles ;
- 5 - Récupérer des secondes informations connexes à une commande, lesdites secondes informations comprenant des données de référence indiquant une liste d'arguments pour ladite commande ;
- Générer de la documentation associée à ladite commande en récupérant un sous-ensemble desdites premières informations déterminé selon ladite liste 10 d'arguments, et fusionner au moins une partie desdites secondes informations avec ledit sous-ensemble desdites premières informations.

Les modes de réalisation préférés comprennent une ou plusieurs des caractéristiques suivantes, lesquelles peuvent être utilisées individuellement ou 15 communément, sous une forme partiellement ou totalement combinée :

- Le programme logiciel est un interpréteur de ligne de commande.
- Lesdites commandes comprennent au moins deux types d'arguments.
- Lesdites premières informations et lesdites secondes informations associent des éléments et des valeurs, et lesdits éléments sont organisés de façon hiérarchique.
- 20 - Un niveau de la hiérarchie desdites premières informations correspond auxdits types d'arguments ;
- Ladite documentation est une page de manuel.

D'autres caractéristiques et avantages de l'invention apparaîtront à la lumière de la description ci-après des modes de réalisation de l'invention, donnés à titre 25 d'exemple non limitatif, en référence aux dessins annexés répertoriés ci-dessous.

## **BRÈVE DESCRIPTION DES DESSINS**

La figure 1 représente un schéma fonctionnel de haut niveau d'un mode de 30 réalisation de l'invention.

## **DESCRIPTION DÉTAILLÉE DE L'INVENTION**

Selon la présente invention, le processus de génération de la documentation 35 associée à un programme logiciel est automatisé afin que tous les éléments de documentation traités deviennent uniformes et qu'aucun élément ne puisse être laissé non rédigé.

Le programme logiciel peut par exemple être un interpréteur de ligne de commande (CLI). De tels interpréteurs sont disponibles dans la plupart des systèmes d'exploitation (OS) et fournissent un accès de bas niveau aux fonctions des systèmes d'exploitation et autres programmes de systèmes. Sur les systèmes Unix ou de type Unix (un système Linux, par exemple), ce type d'interpréteur de ligne de commande peut être appelé « système essentiel ».

L'invention peut également s'appliquer aux programmes logiciels présentant des interfaces utilisateurs graphiques (GUI). En outre, certaines interfaces GUI peuvent être fournies outre l'interpréteur de ligne de commande afin d'en faciliter l'accès. L'exigence de disposer de la documentation pour les commandes disponibles reste la même, mais la documentation générée peut être dans un format plus graphique, par exemple au format HTML ou au format d'aide Windows.

Les interpréteurs de ligne de commande sont dotés de moyens d'analyse permettant aux utilisateurs de saisir des commandes. Les commandes comprennent un nom de commande qui identifie la commande, et des arguments.

Le format des arguments dépend du système d'exploitation, sur l'interpréteur de ligne de commande et de la commande elle-même. La documentation associée à l'interpréteur de ligne de commande a pour but d'expliquer l'utilisation des commandes disponibles.

Un exemple de format possible d'une telle commande peut être :

CMD [<global options>] <action> <types> [<action-options>] <parameters>  
(CMD [<options globales>] <action> <types> [<options d'actions>]  
<paramètres>)

« CMD » représente le nom ou l'identifiant de la commande. Il s'agit d'une chaîne de caractères qui est interprétée par l'interpréteur de ligne de commande en vue de déterminer un traitement particulier à initier, par le biais d'un module particulier de l'interpréteur de ligne de commande ou par le biais d'un programme externe. Certaines commandes sont fournies avec les systèmes d'exploitation et/ou l'interpréteur de ligne de commande, d'autres peuvent être ajoutées librement à un moment ultérieur.

Un tel nom de commande peut être par exemple « cdbm-cluster », lequel représente une commande de gestion de base de données.

« [<global options>] <action> <types> [<action-options>] <parameters> » représente un ensemble d'arguments.

Les arguments peuvent être de différents types. Ils peuvent être divisés en deux types principaux :

- « actions » ; et
- « options », dans lequel peut également être introduit le type « types ».

Les actions sont destinées à spécifier une action liée à la commande. Dans l'exemple de la commande de gestion de base de données « cdbm-cluster », de telles actions peuvent représenter les actions possibles sur une base de données. Elles 5 peuvent être représentatives, par exemple, des actions typiques telles que celles spécifiées dans la base de données SQL (Structure Query Language), normalisées par l'Organisation internationale de normalisation (ISO) et par l'American National Standards Institute (ANSI). Plus précisément, les actions possibles peuvent être, par exemple, « Add » (« ajouter »), « Delete » (« supprimer »), « Commit » 10 (« valider »), « Locate » (« localiser ») ...).

L'argument « options » peut être utilisé en vue de fournir des informations supplémentaires sur la façon dont l'action doit être mise en œuvre. Certaines « options » sont appelées « options globales » parce qu'elles s'appliquent à toutes les actions. Certaines « options » sont appelées « options d'actions », car elles sont 15 spécifiques à une action particulière.

Les « options globales » et « options d'actions » peuvent être différencierées par le moyen d'analyse du programme logiciel (par exemple, l'interpréteur CLI) en fonction de l'ordre des arguments, mais également par des « caractères d'échappement » ou commandes d'interruption spécifiques, par exemple « -- », qui 20 introduisent des options globales.

La table ci-dessous présente des exemples d'options globales pour la commande « cdbm-cluster » :

25	--help	-h :	Affichage de l'aide
	--Debug	-D :	Mode débogage (affichage de tous les niveaux de messages de trace en sortie)
	--quiet	-q :	Mode silencieux (affichage des messages de niveau d'erreur « <i>ERROR</i> » en sortie uniquement)
30	--verbose	-v :	Mode prolix (affichage de tous les niveaux de messages de trace en sortie, hormis le niveau débogage « DEBUG »)
	--noexec	:	Aucune exécution. Simple affichage de la requête exécutée.
	--version	:	Affichage de la version de la commande
35	--base	-b :	Nom de la base de données
	--dbhost	:	Hôte de la base de données
	--dbport	:	Port de la base de données
	--dbuser	:	Utilisateur de la base de données

--dbpassword : Mot de passe de la base de données

La table ci-dessous présente des exemples d'actions pour la commande « cdbm-cluster » :

5

	<i>show</i>	:	Affichage des données de la base de données de grappe « clusterDB » (données de base par défaut)
	<i>add</i>	:	Ajout de données dans clusterDB
10	<i>update</i>	:	<i>Mise à jour de données dans clusterDB</i>
	<i>delete</i>	:	<i>Suppression de données dans clusterDB</i>
	<i>generate</i>	:	<i>Génération d'un fichier 'json'</i>
	<i>rename</i>	:	Renommer une plage de nœuds caractérisés par le préfixe [i-j] en utilisant un nouveau préfixe [m-n].

15

« ClusterDB » correspond au nom de la base de données par défaut gérée par la commande « cdbm-cluster ».

Les « types » sont assez semblables aux « options d'actions » étant donné qu'ils sont spécifiques à une action et donnent des indications supplémentaires sur la façon 20 d'exécuter l'action.

L'invention propose de générer la documentation associée au programme logiciel, par exemple, à l'interpréteur CLI.

La figure 1 montre un schéma de haut niveau décrivant un mode de réalisation 25 de l'invention.

Selon ce mode de réalisation, un dispositif de génération de documentation DG est fourni. Ce dispositif de génération de documentation DG peut être un élément de programme logiciel accessible, à la manière d'une commande, à partir de l'interpréteur de ligne de commande CLI. L'élément peut présenter lui-même un 30 nom de commande et comporter des arguments.

Les entrées du dispositif de génération de documentation DG sont fournies par un système de fichiers FS. Ce système de fichiers FS peut être structuré de différentes façons, et peut être déployé dans le disque dur de l'ordinateur sur lequel le 35 dispositif de génération de documentation DG est exécuté, ou sur un ordinateur distant ou un serveur accessible par des moyens de mise en réseau. Il peut également être téléchargé dans la mémoire vive (RAM) lors de l'accès par le dispositif de génération de documentation DG.

Tel que cela sera expliqué ci-dessous, le système de fichiers FS comporte au moins des premières informations, AI, et des secondes informations, CI. Ces premières et secondes informations peuvent être incluses dans un même fichier ou dans des fichiers distincts. Par exemple, un fichier distinct comprenant les secondes 5 informations peut être fourni pour chaque commande. Les premières informations connexes aux arguments peuvent être réparties sur les fichiers connexes aux « options globales », sur les fichiers connexes aux « options d'actions » pour chaque action, etc.

De nombreux modes de réalisation sont possibles et accessibles pour l'homme 10 du métier. Dans la suite du présent document, le système de fichiers FS est considéré comme une structure de données générique, indépendante de son mode de réalisation spécifique.

Les sorties du dispositif de génération de documentation DG sont des fichiers 15 de documentation ou des « pages de manuel » (« man pages » en anglais), selon la terminologie usuelle du domaine.

Les premières informations AI sont connexes à un ensemble d'arguments possibles qui peuvent être utilisés par les différentes commandes disponibles sur le 20 programme logiciel (ou interpréteur CLI).

Par exemple, lesdites premières informations AI peuvent inclure les données suivantes :

option.debug.help.msg = mode de débogage (affichage de tous les niveaux de messages de trace en sortie)  
25 option.debug.help.long = Debug  
option.debug.help.short = D  
option.debug.help.man = le mode de débogage peut être utilisé pour vérifier ce qui est mis en œuvre à l'intérieur du programme. Il permet de retrouver précisément un problème.  
30 action.show.help.usage=[<global-options>] show <types> [--getid] [--delimiter] [--noheader] [--columns <column>] [--filter <field=value> ] [--order <values>]

Ces données comprennent des premières informations connexes à l'option de 35 débogage « debug » et des premières informations connexes à l'action d'affichage « show ».

Elles traduisent une structure de données spécifique associée à un mode de réalisation de l'invention. Chaque fragment d'information associe un élément et une valeur. Les éléments sont organisés de manière hiérarchique.

Les termes « option » et « action » représentent des clés indiquant que 5 l'élément se rapporte à une option, et respectivement à une action. En d'autres termes, un niveau de la hiérarchie (ici le premier niveau) pour ces premières informations AI, correspond aux types d'arguments (« option », « action » ...).

Le terme « debug » (débogage) est le nom de l'option envisagée.

10 Le terme « help » (aide) signifie que l'élément est afférent à la génération de documentation. La même structure de données peut également être utilisée à d'autres fins. Lorsque seule la génération de documentation est envisagée, ce terme peut être supprimé de la hiérarchie.

15 Les termes « msg », « long », « short », « man » se réfèrent à différents éléments qui peuvent être utilisés afin de générer des fragments de documentation pour l'option en question. En règle générale, ces termes sont normalisés, ou ils correspondent à un usage commun dans le domaine, de manière à fournir un format conventionnel pour la documentation finale générée.

- 20
- En particulier, « msg » indique un message d'explication destiné à être affiché dans la documentation générée.
  - « long » se rapporte à la longue expression destinée à invoquer l'option (ici « -Debug ».)
  - « short » se réfère à l'expression courte destinée à invoquer l'option (ici « -D »)
  - « man » se réfère à une explication plus longue (par rapport à « msg ») destinée à être affichée sous la forme d'une explication complète de la documentation finale générée.
- 25

Bien sûr, d'autres termes peuvent être définis, en formant des éléments de 30 documentation correspondants pour les options.

Les arguments « types », « actions », « options d'actions » et potentiellement d'autres types d'arguments peuvent être traités de la même façon.

Par exemple, l'élément ci-après décrit l'utilisation d'une action d'affichage 35 « show ». De même, d'autres éléments peuvent être fournis pour l'action « show », ainsi que pour d'autres actions.

Son utilisation correspond à un texte destiné à être affiché dans la documentation générée, en vue d'indiquer à l'utilisateur potentiel comment l'action doit être utilisée et en particulier son format.

5           action.show.help.usage=[<global-options>] show <types> [--getid] [--  
delimiter] [--noheader] [--columns <column>] [--filter <field=value> ] [--order  
<values>]

Les secondes informations CI contenues dans le système de fichiers FS se réfèrent à une commande.

10         Elles peuvent comprendre des fragments d'information destinés à être affichés sur la documentation générée, MP. Ces fragments d'information peuvent être, par exemple, une description de la commande, de son utilisation, etc.

Par exemple, ces secondes informations peuvent comprendre l'élément suivant :

15         mycommand.description = mycommand est une commande de gestion de grappe utilisée pour interagir avec la base de données de grappe clusterDB. [...]

où « mycommand » est un nom associé à cette commande.

20         Le terme « description » se réfère à une description de la commande en texte brut. D'autres éléments, introduits par d'autres termes, peuvent se référer à d'autres aspects de la documentation générée MP attendue. Par exemple: « name » (nom), « synopsis », « configuration », etc.

25         En outre, les secondes informations comportent des données de référence indiquant une liste d'arguments pour ladite commande.

Cette liste d'arguments peut bien entendu comporter des options, des actions, des types ou d'autres types d'arguments, selon le mode de réalisation.

Par exemple, les données de référence peuvent se présenter sous la forme ci-dessous :

```
mycommand.option.global=help,debug,quiet,verbose,noexec  
mycommand.option.action.show=delimiter,noheader,columns,filter,order
```

35         Selon cet exemple, le premier élément signifie que les options globales « help », « debug », « quiet », « verbose » et « noexec » sont disponibles pour la commande « mycommand ». Les significations de ces options globales ont été données précédemment.

Le second élément signifie que les options spécifiques à l'action « show » sont « delimiter » ( séparateur), « noheader » (pas d'entête), « columns » (colonnes),

« filter » (filtre) et « order » (ordre). Les significations de ces options ont été données précédemment.

En outre, les actions, les types, et éventuellement d'autres types d'arguments peuvent être spécifiés de la même façon.

5

Selon ces premières et secondes informations fournies par le système de fichiers FS, le dispositif de génération de documentation DG peut générer la documentation associée à la commande « mycommand ».

Cette documentation fait partie de la documentation associée au programme logiciel (par exemple, l'interpréteur CLI). En d'autres termes, le regroupement des documentations générées pour chaque commande fournira la documentation complète du programme logiciel. L'un des objectifs recherchés par l'invention consiste à faciliter la génération de la documentation complète, en l'automatisant et en regroupant les parties communes de la documentation de chaque commande du programme logiciel.

La génération peut être mise en œuvre en récupérant dans un premier temps un sous-ensemble des premières informations AI. Ce sous-ensemble est déterminé selon la liste d'arguments indiqués par les données de référence incluses dans les secondes informations CI.

20 Par exemple, si les données de référence sont données comme suit :

```
mycommand.option.global=help,debug,quiet,verbose,noexec
mycommand.option.action.show=delimiter,noheader,columns,filter,order
```

Alors, les premières informations AI correspondant aux arguments « help », « debug », « quiet », « verbose » et « noexec », « delimiter », « noheader », 25 « columns », « filter » et « order » sont récupérées à partir du système de fichiers FS.

Ensuite, la génération consiste à fusionner au moins une partie des secondes informations CI avec cette extraction du sous-ensemble des premières informations AI.

30 En d'autres termes, les premières informations peuvent être écrites une seule fois, et chaque nouvelle commande ne doit spécifier que les données de référence appropriées (pointant vers les arguments pertinents, qu'il s'agisse d'actions ou d'options, ou de types, etc.), et le dispositif de génération de documentation DG peut réutiliser automatiquement les premières informations disponibles en vue d'élaborer une documentation comprenant à la fois des informations dédiées à la commande 35 (autrement dit les secondes informations CI), et des informations connexes aux différents arguments (autrement dit les premières informations AI).

De même, si une modification doit être apportée à la documentation connexe à un argument, l'exécution du dispositif de génération de documentation DG est

suffisante pour s'assurer que cette modification se reflète automatiquement dans toutes les parties de documentation des commandes utilisant cet argument.

Il n'est par conséquent plus nécessaire de modifier manuellement ces parties de la documentation. Cette tâche étant fastidieuse et source d'erreurs, dans la pratique 5 elle n'était souvent pas effectuée, ou ne l'était que partiellement. Grâce à l'automatisation et à l'utilisation mutuelle des informations communes, l'invention garantit que la documentation disponible est toujours présente et actualisée.

Ci-dessous est présenté un exemple d'une éventuelle documentation générée, 10 dans la forme où celle-ci aurait pu été fournie par un mode de réalisation de l'invention. L'exemple concerne une commande « cdbm-cluster », et fait référence à des « actions » et « options » qui ont déjà été mentionnées dans le présent document.

Dans cet exemple, la documentation générée est du type « pages de manuel ». D'autres formats sont possibles, par exemple le format HTML (langage de balisage 15 hypertexte), format d'aide Windows (“.hlp”), format de document Windows (“.doc”), etc. Certains de ces formats peuvent être adaptés afin d'être affichés sur un écran, mais d'autres peuvent être pertinents ou adaptés à la fourniture d'une documentation imprimée. Le générateur de documentation peut générer différents types de formats de documentation.

20

**cdbm-cluster(1) Cluster Management Commands Manual cdbm-cluster(1)**

## NOM

cdbm-cluster - Commande de gestion de bases de données.

25

## SYNOPSIS

```
cdbm-cluster  [<global-options>]  <action>  <types>  [<action-options>]
<parameters>
```

## 30 CONFIGURATION

Une configuration est déjà définie lors de l'installation du progiciel. Vous pouvez modifier cette configuration dans les fichiers situés dans le dossier /BCM/conf/clusterdb (reportez-vous à la section « FILES »).

## 35 DESCRIPTION

« cdbm-cluster » est une commande de gestion de grappe utilisée pour interagir avec la base de données clusterDB.

Elle gère tous les objets liés à la partie grappe :

- « cluster » : toutes les informations concernant la grappe
  - « cluster profil » : le profil de la grappe (les services installés et leur état)
  - « cluster status » : l'état de la grappe
  - « clusterdb log » : toutes les modifications importantes effectuées sur la base de données clusterDB (application de correctifs, corrections de bogues ...)
- 5

## OPTIONS GLOBALES

Les arguments obligatoires pour les options longues sont obligatoires pour les options courtes également.

- |    |  |
|----|--|
| 10 | <b>-h, --help</b><br>Affichage de l'aide   |
|    | <b>-D, --debug</b><br>Mode débogage (affichage de tous les niveaux de messages de trace en sortie)   |
|    | <b>-q, --quiet</b><br>Mode silencieux (affichage des messages de niveau d'erreur « <i>ERROR</i> » en sortie uniquement)                                    |
| 15 | <b>-v, --verbose</b><br>Mode prolix (affichage de tous les niveaux de messages de trace en sortie, hormis le niveau débogage « <i>DEBUG</i> »)             |
| 20 | <b>--noexec</b><br>Aucune exécution. Simple affichage de la requête exécutée.  |
|    | <b>--version</b><br>Affichage de la version de la commande   |
| 25 | <b>-b, --base[=DATABASE]</b><br>Nom de la base de données (à remplacer par la valeur par défaut lire dans le dossier /BCM/conf/clusterdb/clusterdb.conf)   |
|    | <b>--dbhost[=HOST]</b><br>Hôte de la base de données (à remplacer par la valeur par défaut lire dans le dossier /BCM/conf/clusterdb/clusterdb.conf)        |
| 30 | <b>--dbport[=PORT]</b><br>Port de la base de données (à remplacer par la valeur par défaut lire dans le dossier /BCM/conf/clusterdb/clusterdb.conf)        |
|    | <b>--dbuser[=USER]</b><br>Utilisateur de la base de données (à remplacer par la valeur par défaut lire dans le dossier /BCM/conf/clusterdb/clusterdb.conf) |
| 35 | <b>--dbpassword[=PASSWORD]</b>   |

Mot de passe de la base de données (à remplacer par la valeur par défaut lue dans le dossier /BCM/conf/clusterdb/clusterdb.conf)

#### TYPES

- 5 Les types sont obligatoires pour toutes les actions, hormis pour l'action d'affichage « show » dont la valeur par défaut correspond aux données de base (les autres n'ont pas de type par défaut).

- all « basic », « log », « profile » et « status data ».
- 10 basic Données de base concernant la grappe.
- log Journalisation des modifications apportées à la base de données clusterDB.
- profile Profil de la grappe.
- status État de la grappe.

#### 15 ACTIONS

##### « show »

Affichage des données de la base de données clusterDB (données de base par défaut)

- 20 Avertissement : le type « all » (toutes/tous) n'est pas disponible pour l'action d'affichage « show ».

##### --delimiter[=DELIMITER]

Caractère séparateur utilisé pour délimiter la sortie de données au lieu du caractère d'échappement « space char »

##### --noheader

- 25 Pour supprimer un entête avec des noms de colonnes

##### --columns[=COLUMNS]

Pour sélectionner les colonnes à afficher. Elles doivent être séparées par une virgule.

Exemple : node,interface

- 30 Si vous ajouter le signe '+,' au début de « --columns », l'affichage normal du type choisi sera appliqué et les valeurs que vous avez choisies seront ajoutées.

##### --filter

Filtre pour l'affichage. Forme : <field>=<value> ou

- 35 <table>.<field>=<value>

##### --order

Pour définir l'ordre de l'affichage

L'invention a été décrite en référence à des modes de réalisation préférés. Cependant, de nombreuses modifications et variations sont possibles sans s'écartez du champ d'application de l'invention.

**REVENDICATIONS**

1. Procédé de génération automatique de la documentation associée à un programme logiciel présentant un moyen d'analyse permettant de saisir des commandes comportant des arguments, comprenant les étapes ci-dessous consistant à :

- Fournir des premières informations (AI) connexes à un ensemble d'arguments possibles ;
- Fournir des secondes informations (CI) connexes à une commande, lesdites secondes informations comprenant des données de référence indiquant une liste d'arguments pour ladite commande ;
- Générer de la documentation associée à ladite commande en récupérant un sous-ensemble desdites premières informations déterminé selon ladite liste d'arguments, et fusionner au moins une partie desdites secondes informations avec ledit sous-ensemble desdites premières informations.

2. Procédé selon la revendication 1, dans lequel ledit programme logiciel est un interpréteur de ligne de commande.

3. Procédé selon l'une quelconque des revendications précédentes, dans lequel lesdites commandes comprennent au moins deux types d'arguments.

4. Procédé selon la revendication 3, dans lequel lesdites premières informations et lesdites secondes informations associent des éléments et des valeurs, et dans lequel lesdits éléments sont organisés de façon hiérarchique.

5. Procédé selon la revendication 4, dans lequel un niveau de la hiérarchie pour lesdites premières informations (IA) correspond auxdits types d'arguments.

6. Procédé selon l'une quelconque des revendications précédentes, dans lequel ladite documentation est une page de manuel.

7. Dispositif de génération de documentation (DG) apte à générer de la documentation associée à un programme logiciel présentant un moyen d'analyse permettant de saisir des commandes comportant des arguments, ledit dispositif de génération de documentation comprenant un moyen pour

- Récupérer des premières informations (AI) connexes à un ensemble d'arguments possibles ;
  - Récupérer des secondes informations (CI) connexes à une commande, lesdites secondes informations comprenant des données de référence indiquant une liste d'arguments pour ladite commande ;
  - Générer de la documentation associée à ladite commande en récupérant un sous-ensemble desdites premières informations, déterminé selon ladite liste d'arguments, et fusionner au moins une partie desdites secondes informations avec ledit sous-ensemble desdites premières informations.
- 10            8. Dispositif de génération de documentation (DG) selon la revendication 7, dans lequel ledit programme logiciel est un interpréteur de ligne de commande.
- 15            9. Dispositif de génération de documentation (DG) selon l'une quelconque des revendications 7 et 8, dans lequel lesdites commandes comprennent au moins deux types d'arguments.
- 20            10. Dispositif de génération de documentation (DG) selon la revendication 9, dans lequel lesdites premières informations et lesdites secondes informations associent des éléments et des valeurs, et dans lequel lesdits éléments sont organisés de façon hiérarchique.
- 25            11. Dispositif de génération de documentation (DG) selon la revendication 10, dans lequel un niveau de la hiérarchie pour lesdites premières informations (AI) correspond auxdits types d'arguments.
12. Dispositif de génération de documentation (DG) selon l'une quelconque des revendications 7 à 11, dans lequel ledit document est une page de manuel.

**RAPPORT DE RECHERCHE  
PRÉLIMINAIRE**

établi sur la base des dernières revendications  
déposées avant le commencement de la recherche

N° d'enregistrement  
nationalFA 781527  
FR 1353213

<b>DOCUMENTS CONSIDÉRÉS COMME PERTINENTS</b>		Revendication(s) concernée(s)	Classement attribué à l'invention par l'INPI
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes		
X	WO 2005/029324 A2 (ELECTRONIC DATA SYST CORP [US]; KAMALAKANTHA CHANDRA [US]) 31 mars 2005 (2005-03-31) * page 5, ligne 30 - page 7, ligne 20 * * page 10, ligne 31 - page 12, ligne 2 * * figures 1,4 * -----	1-12	G06F17/24 G06Q10/00
X	"clig - generate a command line parser and/or basic manual page", Ubuntu Manuals, 1 janvier 2010 (2010-01-01), pages 1-4, XP055093083, Extrait de l'Internet: URL: <a href="http://manpages.ubuntu.com/manpages/gutsy/man1/clig.1.html#contenttoc0">http://manpages.ubuntu.com/manpages/gutsy/man1/clig.1.html#contenttoc0</a> [extrait le 2013-12-12] * le document en entier * -----	1-12	
A	"Command-line interface", , 7 avril 2013 (2013-04-07), pages 1-11, XP055093093, Extrait de l'Internet: URL: <a href="http://en.wikipedia.org/w/index.php?title=Command-line_interface&amp;oldid=549228046">http://en.wikipedia.org/w/index.php?title=Command-line_interface&amp;oldid=549228046</a> [extrait le 2013-12-12] * Sections "Operating system command-line interfaces", page 1, "Anatomy of a shell CLI", pages 3 to 7: command prompt, arguments, command-line option, built-in usage help, "Command-line interpreter", pages 7 to 8 * -----	1-12	DOMAINES TECHNIQUES RECHERCHÉS (IPC)
			G06F G06Q
1	Date d'achèvement de la recherche  12 décembre 2013	Examinateur  Glaser, Norbert	
CATÉGORIE DES DOCUMENTS CITÉS		T : théorie ou principe à la base de l'invention E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure. D : cité dans la demande L : cité pour d'autres raisons ..... & : membre de la même famille, document correspondant	
X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : arrière-plan technologique O : divulgation non-écrite P : document intercalaire			

**ANNEXE AU RAPPORT DE RECHERCHE PRÉLIMINAIRE  
RELATIF A LA DEMANDE DE BREVET FRANÇAIS NO. FR 1353213 FA 781527**

La présente annexe indique les membres de la famille de brevets relatifs aux documents brevets cités dans le rapport de recherche préliminaire visé ci-dessus.

Les dits membres sont contenus au fichier informatique de l'Office européen des brevets à la date du **12-12-2013**

Les renseignements fournis sont donnés à titre indicatif et n'engagent pas la responsabilité de l'Office européen des brevets, ni de l'Administration française

Document brevet cité au rapport de recherche	Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
WO 2005029324 A2	31-03-2005	AU 2004273930 A1 CA 2537312 A1 EP 1665035 A2 US 2005060688 A1 WO 2005029324 A2	31-03-2005 31-03-2005 07-06-2006 17-03-2005 31-03-2005