

12

DEMANDE DE BREVET D'INVENTION

A1

22 Date de dépôt : 04.11.20.

30 Priorité :

71 Demandeur(s) : VITESCO TECHNOLOGIES GmbH—
DE.

72 Inventeur(s) : MORYC Roman et ROMEA Nicolas.

43 Date de mise à la disposition du public de la
demande : 06.05.22 Bulletin 22/18.

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 :

73 Titulaire(s) : VITESCO TECHNOLOGIES GmbH.

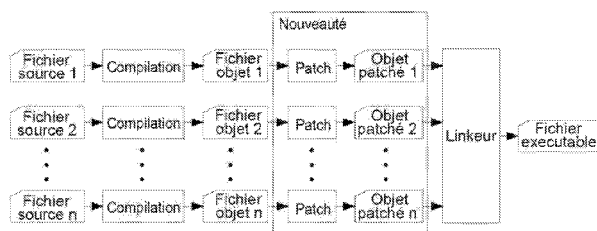
○ Demande(s) d'extension :

74 Mandataire(s) : VITESCO TECHNOLOGIES
FRANCE.

54 Procédé de vérification de compatibilité des interfaces d'un applicatif logiciel.

57 Procédé de vérification de compatibilité des interfaces
d'un applicatif logiciel, comprenant les étapes suivantes :-
compilation des fichiers sources pour produire des fichiers
objets, - modification, dans les fichiers objets, des noms des
objets logiciels, pour leur adjoindre un numéro de version, -
édition des liens.

Figure d'abrégé: Figure 2



Description

Titre de l'invention : Procédé de vérification de compatibilité des interfaces d'un applicatif logiciel

Domaine technique

[0001] L'invention concerne le domaine de la production d'applicatif logiciel et plus particulièrement un procédé de vérification de la cohérence des interfaces lors d'une telle production.

Technique antérieure

[0002] Il est connu pour produire un applicatif logiciel d'utiliser un langage de programmation, tel C ou C++. A l'aide de ce langage, l'applicatif logiciel est écrit dans un ou plusieurs fichiers sources, intelligibles pour un être humain. Au moyen d'un convertisseur nommé compilateur, spécifique du langage de programmation, chaque tel fichier source est converti en un fichier objet, apte à être exécuté par un processeur. Ensuite une édition de liens est réalisée au moyen d'un éditeur de lien ou « linker » qui permet de relier les différents fichiers objets en un unique applicatif logiciel, ou fichier exécutable.

Résumé de l'invention

[0003] Selon l'invention, il est ajouté une étape dans laquelle chaque objet logiciel, que ce soit une donnée ou une routine (fonction ou procédure) est documenté en lui adjoignant une version. Cette version est ajoutée dans la distribution logicielle en remplaçant le nom de chaque objet logiciel « X » par un nom et une version accolée.

[0004] L'invention a pour objet un procédé de vérification de compatibilité des interfaces d'un applicatif logiciel, comprenant les étapes suivantes :

- compilation des fichiers sources pour produire des fichiers objets,
- modification, dans les fichiers objets, des noms des objets logiciels, pour leur adjoindre un numéro de version,
- édition des liens.

[0005] Des caractéristiques ou des modes de réalisation particuliers, utilisables seuls ou en combinaison, sont :

- la modification des noms des objets logiciels, remplace un nom « X » par le même nom accolé d'un symbole de version « _V » et d'un numéro « n » de version, soit « X_Vn »,
- la modification des noms des objets logiciels est réalisée au moyen d'un outil de patch des fichiers objets, tel OBJCOPY,
- la modification des noms des objets logiciels est réalisée au moyen d'un fichier de commande.

Brève description des dessins

- [0006] L'invention sera mieux comprise à la lecture de la description qui suit, faite uniquement à titre d'exemple, et en référence aux figures en annexe dans lesquelles :
- [0007] [fig.1] illustre un procédé de production d'un applicatif logiciel selon l'art antérieur,
- [0008] [fig.2] illustre un procédé de production d'un applicatif logiciel modifié par l'invention,
- [0009] [fig.3] illustre un problème d'interface non détecté selon l'art antérieur,
- [0010] [fig.4] illustre la détection du problème d'interface par l'invention,

Description des modes de réalisation

- [0011] En référence à la [fig.1], un logiciel applicatif se construit en connectant des modules logiciels entre eux. Un applicatif logiciel est compris dans n fichiers sources, ici numéroté de 1..n. La compilation produit autant de fichiers objets. Les modules d'un logiciel applicatif sont très souvent des modules génériques, développés dans l'optique de favoriser leur réutilisation sur de nombreux projets afin de limiter les coûts de développement des projets tout en garantissant un meilleur niveau de qualité et de robustesse.
- [0012] Chaque module possède des interfaces d'entrée (interfaces produites par d'autres modules) et des interfaces de sorties (interfaces produites par ce module pour être utilisée par d'autres modules) lui permettant de communiquer avec les autres modules.
- [0013] Deux méthodes de communication existent :
- Méthode de type < control-flow >): un module appelle une fonction d'un autre module. Une interface de type < control-flow >> est implémentée en définissant une fonction pouvant être appelée depuis plusieurs modules,
 - Méthode de type < data-flow > : un module lit une donnée écrite par un autre module. Une interface de type < data-flow > est implémentée en définissant une donnée en mémoire globale partagée pouvant être lue et écrite par plusieurs modules.
- [0014] Les développements de modules génériques sont effectués au sein de plates-formes logicielles définissant les interfaces standards ce qui facilite l'intégration des modules. Les interfaces sont généralement définies de manière centralisée, dans une base de données, telle qu'illustrée à la [fig.3] ou 4, dans laquelle chaque interface est décrite. Par exemple, une donnée possède un nom, un type et une unité.
- [0015] Selon une caractéristique de l'invention, les interfaces sont versionnées afin de gérer les changements : à chaque modification d'une interface, sa version est modifiée.
- [0016] A titre d'exemple, un objet tension batterie est présenté. Un tel objet logiciel, nommé VB est une sortie du module qui gère la tension batterie. Il constitue alors une entrée pour tous les modules qui lisent et/ou utilisent la tension batterie.
- [0017] Il convient que l'interface d'un tel objet VB soit clairement définie de la même

manière pour l'objet qui produit cette valeur et la met à disposition et pour tous les utilisateurs.

- [0018] On suppose qu'une première version de l'interface de l'objet VB est de type linéaire 8-bit non-signé [0..25,8V] en Volt. Suite à une évolution, une deuxième version de l'interface de l'objet VB est définie de type linéaire 8-bit non-signé [0...28,8V] en Volt. Il peut être vu qu'une différence d'interface peut ici avoir des conséquences importantes. En effet si la valeur 0 correspond dans les deux versions à une valeur 0V, la valeur FF correspond à 25,8V pour la version 1 et à 28,8V pour la version 2, avec des variations plus ou moins sensibles pour toutes les autres valeurs. Aussi, il est de première importance qu'un module utilisateur de l'objet tension batterie VB utilise bien la version d'interface à laquelle il s'attend.
- [0019] Selon l'art antérieur illustré à la [fig.3], tous les objets tension batterie VB, portent le même nom VB et ne peuvent être distingués. Aussi, tel qu'illustré à la [fig.3], qu'il s'agisse d'un VB de version 1 ou de version 2, l'éditeur de liens voit un seul et même nom VB dans les deux cas. Il n'est pas en mesure de détecter une incohérence d'interface.
- [0020] Aussi, selon une étape importante de l'invention, plus particulièrement visible à la [fig.2], il est ajouté une étape de documentation, en introduisant les numéros de version dans les fichiers objets. Pour cela, chaque fichier objet est modifié ou « patché ». Chaque fichier objet est remplacé par un fichier objet « patché » dans lequel tous les noms de tous les objets logiciels sont modifiés. Cette modification permet d'introduire une version associée à chaque objet logiciel. Tous les noms étant modifiés, la modification s'applique tant à un nom utilisé comme producteur que au même nom utilisé comme consommateur. Ceci permet alors de réaliser une vérification des versions des interfaces lors de l'édition de lien ultérieure.
- [0021] C'est l'éditeur de liens, lors de sa tâche d'assemblage qui comporte une vérification de la cohérence des noms des objets logiciels qui réalise la vérification de cohérence des interfaces.
- [0022] Ainsi, tel qu'illustré à la [fig.4], l'objet tension batterie VB selon la version 1 est patché en VB_V1, tandis que l'objet tension batterie VB selon la version 2 est patché en VB_V2. Aussi, il apparaît deux noms différents, puisque correspondant à deux versions d'interfaces, et l'éditeur de liens est en mesure de détecter un problème de différence d'interface.
- [0023] Aussi, le procédé de vérification de compatibilité des interfaces d'un applicatif logiciel, comprend les étapes suivantes :
- compilation des fichiers sources pour produire des fichiers objets,
 - modification, dans les fichiers objets, des noms des objets logiciels, pour leur adjoindre un numéro de version,

- édition des liens.

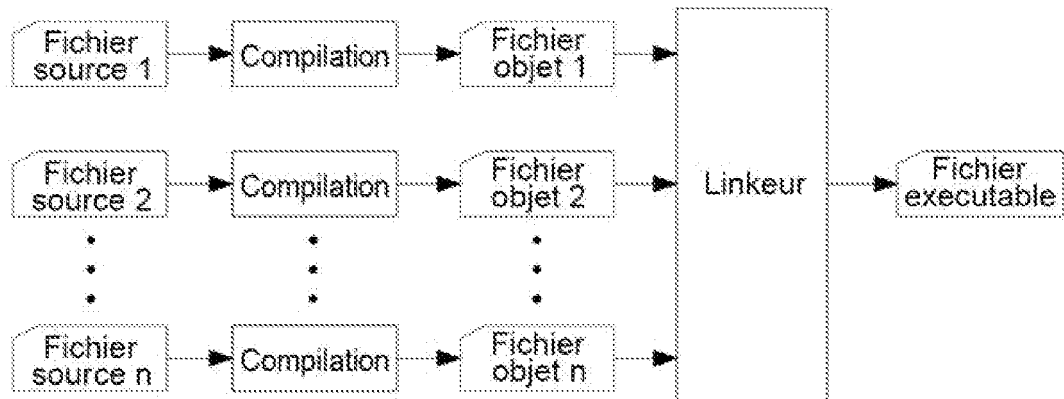
- [0024] Du fait du changement de nom, qui permet de renseigner la version de l'interface, de l'information est introduite dans les fichiers objets. Cette information supplémentaire permet à l'éditeur de liens de vérifier que deux objets logiciels liés présentent bien les bonnes interfaces pour que ce lien soit possible.
- [0025] Selon une autre caractéristique, la modification des noms des objets logiciels, remplace un nom « X » par le même nom accolé d'un symbole de version. Le symbole de version peut être quelconque et par exemple « _V ». Ensuite, un numéro « n » de version peut encore être accolé. Le choix, et l'ordre des différents symboles retenus peuvent être variables. Selon un mode de réalisation, un objet « X » est renommé « X_Vn », avec n le numéro de version.
- [0026] Selon une autre caractéristique, la modification des noms des objets logiciels ou « patch » est réalisée au moyen d'un outil de patch des fichiers objets. Un tel outil de patch existe généralement dans une distribution de développement logiciel.
- [0027] Ce type d'outil est habituellement utilisé pour patcher des symboles dans un code objet dont le code source n'est pas disponible afin de résoudre des problèmes de conflit de nommage des données. Ainsi, par exemple, lorsque les codes objets de deux modules utilisent un même nom pour des données différentes, il faut alors pouvoir renommer le symbole de la donnée dans l'un au moins des modules afin d'éviter un conflit de nommage lors de l'édition de liens. Un tel outil est nommé OBJCOPY dans une distribution C.
- [0028] L'outil de patch peut être utilisé à la main, objet logiciel par objet logiciel. Cependant pour un logiciel d'une certaine taille le procédé devient vite fastidieux et cause d'erreur. Aussi selon une autre caractéristique, un fichier de commande ou script permettant d'automatiser cette tâche est employé. Aussi, la modification des noms des objets logiciels est avantageusement réalisée au moyen d'un fichier de commande.
- [0029] L'invention a été illustrée et décrite en détail dans les dessins et la description précédente. Celle-ci doit être considérée comme illustrative et donnée à titre d'exemple et non comme limitant l'invention à cette seule description. De nombreuses variantes de réalisation sont possibles.

Revendications

- [Revendication 1] Procédé de vérification de compatibilité des interfaces d'un applicatif logiciel, *caractérisé en ce qu'il* comprend les étapes suivantes :
- compilation des fichiers sources pour produire des fichiers objets,
 - modification, dans les fichiers objets, des noms des objets logiciels, pour leur adjoindre un numéro de version,
 - édition des liens.
- [Revendication 2] Procédé selon la revendication 1, où la modification des noms des objets logiciels, remplace un nom « X » par le même nom accolé d'un symbole de version « _V » et d'un numéro « n » de version, soit « X_Vn ».
- [Revendication 3] Procédé selon l'une quelconque des revendications 1 à 2, où la modification des noms des objets logiciels est réalisée au moyen d'un outil de patch des fichiers objets, tel OBJCOPY.
- [Revendication 4] Procédé selon l'une quelconque des revendications 1 à 2, où la modification des noms des objets logiciels est réalisée au moyen d'un fichier de commande.

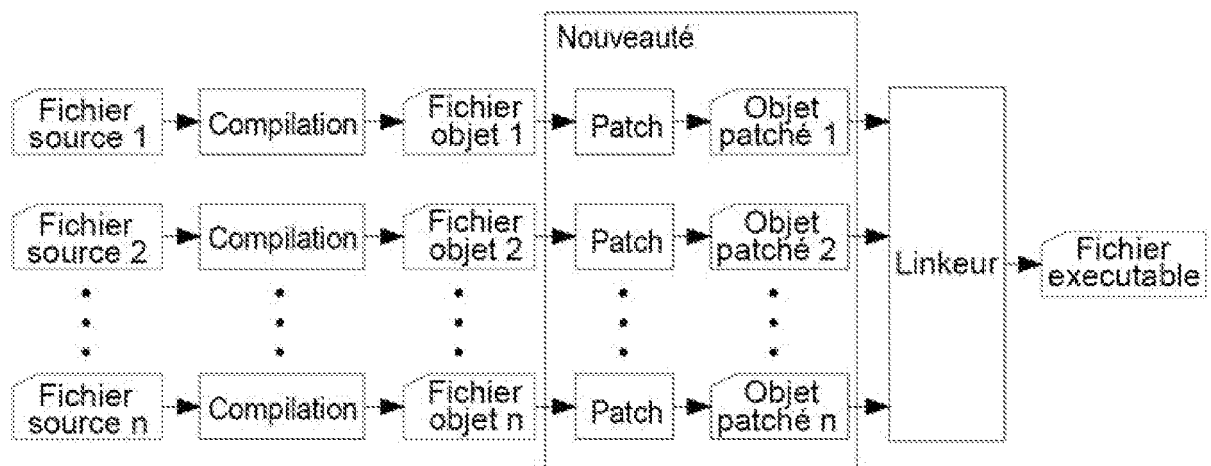
[Fig. 1]

FIG. 1



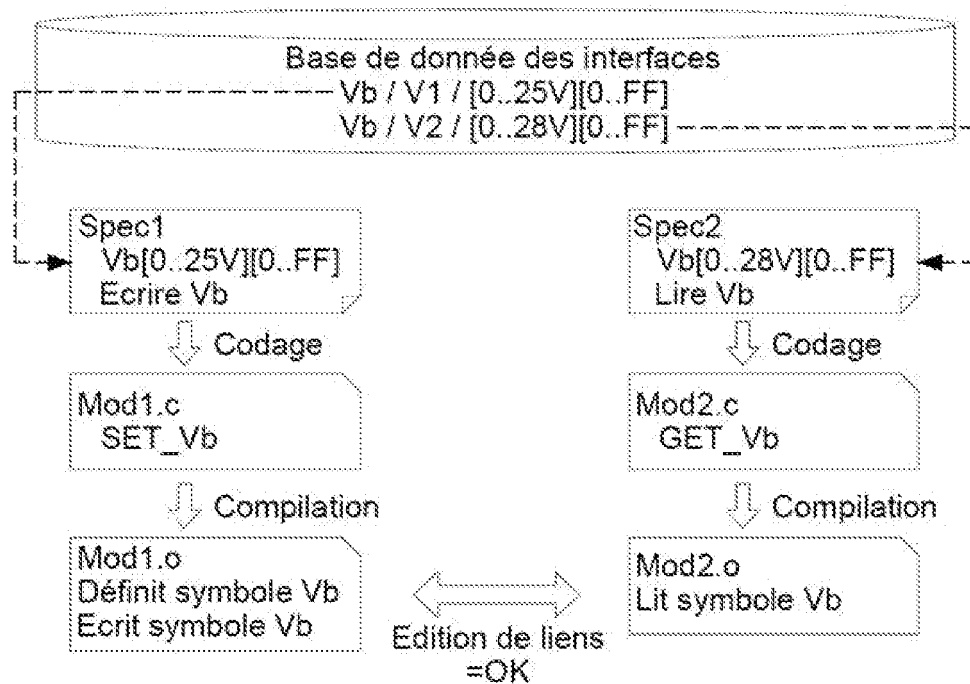
[Fig. 2]

FIG. 2



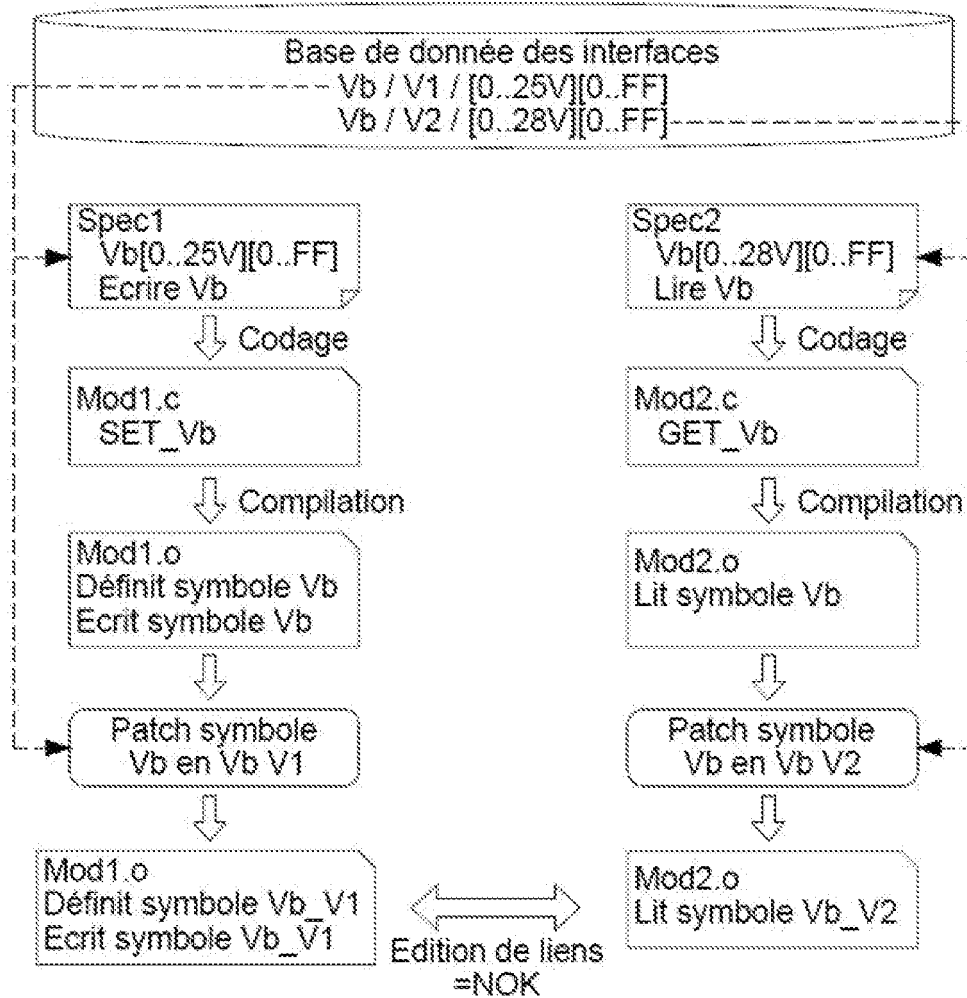
[Fig. 3]

FIG. 3



[Fig. 4]

FIG. 4



RAPPORT DE RECHERCHE PRÉLIMINAIRE

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

N° d'enregistrement
national

FA 888834
FR 2011288

DOCUMENTS CONSIDÉRÉS COMME PERTINENTS		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		
Y	EP 0 833 246 A2 (TEXAS INSTRUMENTS INC [US]) 1 avril 1998 (1998-04-01) * abrégé; revendications 1,2,3 * * page 5, ligne 15 - ligne 28; figure 2(205)(206)(207) * * page 6, ligne 11 - page 7, ligne 12 * -----	1-4	G06F8/54 G06F8/65 G06F8/70
Y	Taylor Ian Lance: "Combining Versions", 18 juillet 2008 (2008-07-18), page 1, XP055829107, Extrait de l'Internet: URL:https://www.airs.com/blog/archives/220 [extrait le 2021-07-30] * le document en entier *	1,2	
Y	Lee Matthew: "Solving C Symbol Collisions", 21 septembre 2017 (2017-09-21), pages 1-3, XP055829194, Extrait de l'Internet: URL:https://www.ivoive.com/solving-c-symbo l-collisions/ [extrait le 2021-07-30] * le document en entier *	3	
A		1	
Y	Pesch Roland H ET AL: "The gnu Binary Utilities Version 2.27", août 2016 (2016-08), pages 1-93, XP055829089, Extrait de l'Internet: URL:https://homepages.uni-regensburg.de/~b rf09510/EDV/kurs_info/brf09510/kurs_info/p df/binutils.pdf [extrait le 2021-07-30] * page 1 - page 4 * * page 15 - page 29 * -----	3,4	
A		1	
Date d'achèvement de la recherche		Examineur	
30 juillet 2021		de Junca, Irène	
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			

RELATIF A LA DEMANDE DE BREVET FRANÇAIS NO. FR 2011288 FA 888834

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 30-07-2021

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	
EP 0833246	A2	01-04-1998	EP 0833246 A2	01-04-1998
			JP H10111807 A	28-04-1998
			US 6002876 A	14-12-1999
