(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

COURBEVOIE

(11) **N° de publication :** **3 031 820**
(à n'utiliser que pour les commandes de reproduction)

(21) **N° d'enregistrement national :** **15 00075**

(51) Int Cl[8] : ***G 06 F 11/36*** *(2017.01), G 06 F 17/50*

(12) BREVET D'INVENTION B1

(54) PROCEDE DE DETERMINATION DE LA VALIDITE D'UN MODELE.

(22) **Date de dépôt :** 15.01.15.

(30) **Priorité :**

(43) **Date de mise à la disposition du public de la demande :** 22.07.16 Bulletin 16/29.

(45) **Date de la mise à disposition du public du brevet d'invention :** 30.03.18 Bulletin 18/13.

(56) **Liste des documents cités dans le rapport de recherche :**

Se reporter à la fin du présent fascicule

(60) **Références à d'autres documents nationaux apparentés :**

◯ **Demande(s) d'extension :**

(71) **Demandeur(s) :** *DASSAULT AVIATION Société anonyme* — FR.

(72) **Inventeur(s) :** CAMPAN CLAIRE et PERFETTO PHILIPPE.

(73) **Titulaire(s) :** DASSAULT AVIATION Société anonyme.

(74) **Mandataire(s) :** CABINET LAVOIX Société par actions simplifiée.

FR 3 031 820 - B1

Procédé de détermination de la validité d'un modèle

La présente invention concerne un procédé mis en œuvre par ordinateur de détermination d'une validité d'un modèle représentatif d'un scénario d'utilisation d'une chaîne fonctionnelle d'un système complexe, le scénario comprenant la mise en œuvre, par au moins un constituant de la chaîne fonctionnelle, d'une séquence d'échanges de messages avec au moins un autre constituant de ladite chaîne fonctionnelle ou un élément extérieur audit système complexe, au moins une séquence d'échanges étant soumise à au moins une contrainte temporelle associée.

L'invention s'applique en particulier à la modélisation de scénarios destinés à être mis en œuvre par un système complexe tel qu'une plate-forme, notamment un aéronef civil ou militaire, ou encore un engin volant, tel qu'un drone. Dans d'autres exemples, la plate-forme est un engin spatial, un engin naval, ou un véhicule terrestre.

Un système complexe comprend un ensemble de constituants reliés par des interconnexions, les constituants et leurs interconnexions définissant l'architecture du système complexe. Cette architecture est hiérarchisée, chaque constituant pouvant être à son tour vu à comme un sous-système et être développé dans un niveau inférieur.

Le fonctionnement de chaque constituant peut être modélisé par un modèle élémentaire. Un modèle global du système complexe peut être obtenu par connexion des modèles élémentaires en fonction des interconnexions définies dans l'architecture du système complexe.

Un tel modèle global s'avère néanmoins excessivement complexe lorsque tous les constituants du système sont intégrés à ce modèle global, ce qui rend la conception du système particulièrement difficile et long.

Pour faciliter cette conception, on a ainsi recours à des modèles partiels chacun spécifique à un scénario d'utilisation du système complexe, par exemple une phase d'initialisation du système complexe. Chaque scénario d'utilisation fait intervenir un ensemble de constituants du système, reliés entre eux conformément à l'architecture prédéfinie du système, et correspond à une chaîne fonctionnelle donnée, c'est-à-dire à un ensemble de sous-fonctions mises en œuvre par un ou plusieurs constituant(s) et contribuant à la réalisation d'une fonctionnalité du système complexe. Chaque scénario peut être modélisé par un ensemble d'échanges entre un ou des constituants de la chaîne fonctionnelle et éventuellement des éléments extérieurs au système complexe, ces échanges étant soumis à des contraintes temporelles.

L'utilisation de tels modèles spécifiques permet notamment de vérifier la cohérence du scénario associé.

Cette vérification est par exemple réalisée au moyen d'outils de modélisation mettant en œuvre le language SysML (pour Systems Modeling Language). De tels outils, s'ils permettent de vérifier la cohérence entre l'architecture du système et les séquences d'échanges du scénario, ne permettent pas de vérifier si le scénario peut effectivement être mis en œuvre, compte tenu des contraintes temporelles associées aux échanges.

On connaît par ailleurs l'outil ROMEO®, destiné à la validation et à la vérification de système temps-réels, et qui permet de vérifier la validité des contraintes de temps dans une architecture système. Cependant, cet outil ne permet pas de prendre en compte une architecture hiérarchisée du système et le lien entre une telle architecture hiérarchisée et l'architecture dynamique liée au scénario.

Un but de l'invention est de fournir un procédé de vérification de la validité d'un scénario d'utilisation d'une chaîne fonctionnelle d'un système complexe qui soit propre à vérifier la cohérence entre l'architecture du système et les séquences d'échanges du scénario, et à vérifier formellement le respect des contraintes temporelles associées à ces échanges.

A cette fin, l'invention a pour objet un procédé du type prédéfini, caractérisé en ce qu'il comprend :

- une étape de fourniture d'un modèle représentatif de l'ensemble des séquences d'échanges destinés à être mise en œuvre par le ou lesdits constituants lors de la mise en œuvre dudit scénario,

- une étape de génération, à partir dudit modèle, d'un modèle à automates temporisés dudit scénario,

- une étape de vérification, à partir dudit modèle à automates temporisés, d'une compatibilité entre les contraintes temporelles associées aux séquences d'échanges dudit scénario, ledit modèle étant considéré comme non valide si au moins deux contraintes temporelles sont incompatibles entre elles.

Le procédé selon l'invention peut comprendre l'une ou plusieurs des caractéristiques suivantes, prise(s) isolément ou suivant toute combinaison techniquement possible :

- l'étape de fourniture dudit modèle comprend :
    - une phase de génération, pour chaque constituant de ladite chaine fonctionnelle, d'un modèle élémentaire représentatif de la séquence d'échanges destinée à être mise en œuvre par ledit constituant et de la ou des contrainte(s) temporelle(s) associée(s) à ladite séquence d'échanges,
    - une phase de détermination dudit modèle représentatif dudit scénario, par composition desdits modèles élémentaires.

- l'étape de génération dudit modèle à automates temporisés comprend la génération, à partir de chacun desdits modèles élémentaires, d'au moins un automate temporisé représentatif de la séquence d'échanges destinée à être mise en œuvre par ledit constituant et de la ou des contrainte(s) temporelle(s) associée(s) à ladite séquence d'échanges, et la génération d'un automate de synchronisation desdits automates temporisés ;

- la génération de chaque automate temporisé comprend la génération d'un automate à états finis représentatif de la séquence d'échanges destinée à être mise en œuvre par ledit constituant, et la génération, pour chacune des contraintes temporelles associées à ladite séquence d'échanges, d'un automate de contrainte représentatif de ladite contrainte temporelle ;

- chaque modèle élémentaire est un modèle élémentaire de type MSC, et ladite phase de détermination dudit modèle représentatif du scénario comprend une composition desdits modèles élémentaires de type MSC, ledit modèle représentatif du scénario étant de type HMSC ;

- la composition desdits modèles MSC élémentaires est réalisée suivant une algèbre de composition comprenant :

  - un premier opérateur, commutatif et associatif,
  - un deuxième opérateur associatif et non commutatif,
  - un troisième opérateur commutatif et non associatif. ;

- chaque contrainte temporelle est associée à un intervalle de temps, à partir d'un événement de référence, dans lequel un événement prédéfini est attendu ;

- chaque contrainte temporelle est associée à un type de vérification choisi parmi :

  - un premier type de vérification, tel que la contrainte temporelle est considérée comme satisfaite si, quel que soit l'instant de l'intervalle de temps auquel ledit événement se produit, les autres contraintes temporelles peuvent également être satisfaites; et
  - un deuxième type de vérification, tel que la contrainte temporelle est considérée comme satisfaite s'il existe au moins un instant de l'intervalle de temps auquel l'événement se produit, permettant aux autres contraintes temporelles d'être satisfaites ;

- lors de ladite étape de vérification, les contraintes temporelles sont jugées compatibles entre elles si lors de la mise en œuvre du scénario, toutes les contraintes temporelles peuvent être satisfaites ;

- ledit procédé comprend la spécification d'au moins une partie des contraintes temporelles, la spécification d'une contrainte temporelle comprenant la définition de la

valeur des bornes inférieure et supérieure de l'intervalle de temps associé à ladite contrainte temporelle ;

- au moins une contrainte temporelle n'est pas spécifiée, et lors de l'étape de vérification, le scénario est considéré comme valide, sous réserve de ladite contrainte temporelle non spécifiée, si toutes les contraintes temporelles spécifiées sont compatibles entre elles ;

- toutes les contraintes temporelles sont spécifiées, et lors de l'étape de vérification, le scénario est considéré comme valide si toutes les contraintes temporelles sont compatibles entre elles ;

- chacun desdits événement de référence et événement prédéfini est une réception ou une émission d'un message par le constituant destiné à mettre en œuvre la séquence d'échanges soumise à ladite contrainte temporelle ;

- chaque contrainte temporelle est d'un type choisi parmi :

  - un premier type de contraintes selon lequel ledit événement de référence et ledit événement prédéfini sont causalement dépendant ;
  - un deuxième type de contraintes destiné à assurer une synchronisation temporelle entre deux constituants ;
  - un troisième type de contraintes destiné à spécifier un intervalle de temps, après ledit événement de référence, dans lequel un constituant est susceptible de prendre en compte un message qu'il reçoit d'un autre constituant ou d'un élément extérieur au système complexe ;
  - un quatrième type de contraintes destiné à spécifier une exigence temporelle garantissant un bon fonctionnement du système.

L'invention a également pour objet un système de détermination d'une validité d'un modèle représentatif d'un scénario d'utilisation d'une chaîne fonctionnelle d'un système complexe, ledit scénario comprenant la mise en œuvre, par au moins un constituant de la chaîne fonctionnelle, d'une séquence d'échanges de messages avec au moins un autre constituant de ladite chaîne fonctionnelle ou un élément extérieur audit système complexe, chaque séquence d'échanges étant soumise à au moins une contrainte temporelle associée,

ledit système étant caractérisé en ce qu'il comprend :

- des moyens de fourniture d'un modèle représentatif de l'ensemble des séquences d'échanges destinés à être mise en œuvre par le ou lesdits constituants lors de la mise en œuvre dudit scénario,

- un élément logiciel de vérification de la validité dudit modèle, ledit élément logiciel de vérification comprenant :

- un module pour générer, à partir dudit modèle, un modèle à automates temporisés dudit scénario,
- un module pour vérifier, à partir dudit modèle à automates temporisés, une compatibilité entre les contraintes temporelles associées aux séquences d'échanges dudit scénario, ledit modèle étant considéré comme non valide si au moins deux contraintes temporelles sont incompatibles entre elles.

L'invention sera mieux comprise à la lecture de la description qui va suivre, donnée uniquement à titre d'exemple, et faite en se référant aux dessins annexés, sur lesquels :

- la Figure 1 est une vue schématique d'un système de vérification selon l'invention destiné à la création et/ou à la modification d'une architecture logique d'un système complexe d'une plate-forme ;

- la Figure 2 est un exemple de représentation graphique de modèles élémentaires associés à des constituants d'un système ;

- la Figure 3 est un exemple de représentation graphique d'un modèle global résultant de la composition des modèles élémentaires de la Figure 2 ;

- les Figures 4 et 5 illustrent deux exemples de représentation graphique d'automates finis générés à partir de modèles élémentaires de la Figure 2 ;

- la Figure 6 illustre un exemple de représentation graphique d'un automate de synchronisation ;

- les Figures 7 et 8 sont des représentations graphiques d'automates de contrainte selon un premier et un deuxième types ;

- la Figure 9 illustre une telle représentation graphique global de la Figure 3

- la Figure 10 est un schéma synoptique illustrant les étapes d'un procédé selon un mode de réalisation de l'invention.

On a illustré sur la Figure 1 un système 10 selon un mode de réalisation de l'invention pour la mise en œuvre d'un procédé de détermination de la validité d'un modèle représentatif d'un scénario d'utilisation d'une chaîne fonctionnelle d'un système complexe.

Le système complexe est un système d'une plate-forme, ou est un système défini par un ensemble de plates-formes interagissant les unes avec les autres.

La plate-forme est par exemple un aéronef civil ou militaire, ou encore un engin volant, tel qu'un drone. Dans d'autres exemples, la plate-forme est un engin spatial, un engin naval, ou un véhicule terrestre.

Le système complexe de la plate-forme est par exemple un système de pilotage de la plate-forme, un système de commande de fonctionnalités d'habitacle de la plate-forme, un système d'actionnement d'éléments mobiles de la plate-forme, ou encore un système de largage d'un objet, tel qu'une arme à partir de la plate-forme.

Le système de détermination 10 comporte un ordinateur 12, comprenant une unité centrale de traitement 14, un écran d'affichage 16 et une interface homme machine 18, par exemple un clavier, un écran tactile, et/ou une souris.

L'unité centrale 14 comporte un processeur 20 et une mémoire 22 contenant un logiciel propre à être exécuté par le processeur 20 pour la mise en œuvre du procédé.

La mémoire 22 contient ainsi un élément logiciel 24 de fourniture d'une architecture hiérarchisée du système complexe, un élément logiciel 26 de fourniture d'un modèle spécifique à un scénario d'utilisation du système complexe.

La mémoire 22 contient en outre un élément logiciel 28 de vérification de la validité d'un modèle spécifique défini pour un scénario d'utilisation donné.

L'architecture hiérarchisée du modèle complexe comprend l'ensemble des constituants du système complexe et les connexions entre ces constituants. Ces connexions sont des liens entre les constituants, qui peuvent être des liens informatifs ou des liens d'énergie (par exemple hydraulique ou électrique). Ces liens permettent la mise en œuvre d'échanges entre les constituants du système complexe.

Les liens informatifs sont dédiés à l'échange d'informations sur l'état d'un paramètre ou de l'un des constituants du système, et à l'acquisition d'informations issues d'interfaces homme/machine du système complexe.

Les liens d'énergie modélisent les ressources physiques nécessaires à la réalisation d'une fonction.

Chaque constituant comprend des interfaces permettant la connexion du constituant aux autres constituants selon l'architecture du système complexe, et éventuellement à un ou des éléments extérieurs au système complexe.

En outre, chaque constituant est associé à une ou plusieurs fonction(s) pouvant être mise en œuvre par ce constituant. Ces fonctions sont généralement mises en œuvre sur la base d'au moins une donnée d'entrée reçue, d'un autre constituant ou d'un élément extérieur au système complexe, et engendrent au moins une donnée de sortie destinée à être transmise à un autre constituant, ou à un élément extérieur au système complexe.

L'élément logiciel 24 est ainsi propre à permettre la création, le chargement, et/ou la modification d'une architecture hiérarchisée du système complexe, incluant les constituants, leurs interfaces, les fonctions associées et les liens entre les constituants.

L'élément logiciel 26 est propre à permettre à permettre la création, le chargement, et/ou la modification, sur la base d'une architecture du système complexe définie au moyen de l'élément logiciel 24, d'un modèle spécifique à un scénario d'utilisation du système complexe.

Un tel scénario fait intervenir un constituant ou un ensemble de constituants du système reliés entre eux conformément à l'architecture prédéfinie du système, et un ensemble de sous-fonctions, formant une chaîne fonctionnelle, mises en œuvre par ce ou ces constituant(s) et contribuant à la réalisation d'une fonctionnalité du système complexe.

Le modèle spécifique au scénario comprend un ensemble d'échanges ordonnancés mis en œuvre par le ou les constituants de la chaîne fonctionnelle.

En particulier, le scénario comprend la mise en œuvre, par le ou chaque constituant de la chaîne fonctionnelle d'une séquence ordonnancée d'échanges de messages avec un ou plusieurs autres constituant(s) de la chaine fonctionnelle ou un élément extérieur au système complexe. Chaque message émis par un constituant fait suite à un message reçu par ce constituant.

Ces séquences d'échanges sont par ailleurs soumises à des contraintes temporelles. Ces contraintes temporelles sont définies par un intervalle de temps, à partir d'un événement de référence tel que la réception ou l'émission d'un message par un constituant, dans lequel un événement prédéfini, tel que la réception ou l'émission d'un autre message par ce constituant, doit être effectué.

Chaque intervalle de temps est défini par une borne inférieure et une borne supérieure. Un intervalle de temps, et la contrainte de temps associée, seront par la suite considérés comme définis si les valeurs de bornes inférieure et supérieure de l'intervalle de temps sont toutes deux définies.

Un modèle est entièrement contraint si toutes les contraintes de temps nécessaires à la vérification du modèle sont spécifiées, ou sous-contraint si au moins une contrainte de temps du modèle n'est pas définie.

La réalisation du scénario nécessite que toutes les contraintes temporelles associées aux séquences d'échanges soient vérifiées, donc que toutes ces contraintes soient compatibles entre elles.

Les contraintes temporelles peuvent être de quatre types donnés.

Un premier type de contraintes comprend les contraintes causales, qui spécifient un délai entre messages causalement dépendants, c'est-à-dire dont l'un est une conséquence de l'autre. Il s'agit par exemple d'un délai dans lequel un constituant émet

un message donné après réception d'un ou plusieurs messages nécessaires à l'élaboration de ce message donné.

Un deuxième type de contraintes comprend les contraintes de synchronisation. Un message est un point de synchronisation entre des constituants d'une chaîne fonctionnelle. Lorsque cette synchronisation concerne plus de deux constituants, il est nécessaire, pour assurer cette synchronisation, de spécifier des contraintes de synchronisation, qui expriment la dépendance temporelle entre des messages modélisant un même point de synchronisation. De telles contraintes de synchronisation sont par exemple utilisées pour assurer une mise sous tension simultanée de constituants d'une chaîne fonctionnelle, en synchronisant les messages de mises sous tension reçus par ces constituants.

Un troisième type de contraintes est une contrainte d'attente, qui spécifie un intervalle de temps dans lequel un constituant récepteur est susceptible de prendre en compte un message qu'il reçoit d'un émetteur. Cet intervalle de temps est défini par rapport à un événement antérieur tel que la réception ou l'émission d'un message antérieur par le constituant. Un point de synchronisation est défini entre l'émetteur et le récepteur.

Un quatrième type de contraintes correspond à des exigences à vérifier pour garantir le bon fonctionnement du système, par exemple un délai entre un message initial et un message final du scénario.

Le modèle spécifique au scénario est ainsi défini par l'architecture statique de l'ensemble de constituants intervenant dans ce scénario, et par une architecture dynamique définie par les contraintes dynamiques du scénario, comprenant les séquences ordonnancées d'échanges pour la mise en œuvre de ce scénario et les contraintes temporelles associées.

Lorsqu'il existe une contrainte de temps entre un échange e_0 à un instant t_0 et un échange e_1 à un instant t_1 tel que $t_1>t_0$, s'il existe au moins un échange e_2 à un instant t_2 tel que $t_1>t_2>t_0$ et que cet ordonnancement des échanges ne peut être garanti compte tenu des contraintes temporelles exprimées, alors un tel modèle est dit sous-contraint. Ceci correspond à une rupture de la chaîne temporelle causale.

Pour définir ce modèle spécifique (éventuellement sous-contraint), l'élément logiciel 26 est propre à générer, pour le ou chaque constituant de la chaine fonctionnelle, un modèle élémentaire, qui est dans l'exemple décrit un modèle MSC (pour « Message Sequence Chart » ou graphe de séquence de messages), représentatif de la séquence d'échanges destinée à être mise en œuvre par le constituant et de la ou des contrainte(s) temporelle(s) associée(s) à cette séquence d'échanges.

A cette fin, l'élément logiciel 26 comprend un éditeur de modèle, qui est dans l'exemple décrit un éditeur MSC, utilisant le langage MSC, propre à générer un modèle MSC élémentaire pour chaque constituant de la chaîne fonctionnelle, par exemple en fonction de données relatives au scénario modélisé stockées dans la mémoire 22 ou saisies par un opérateur via l'interface homme machine 18.

L'éditeur MSC est un éditeur graphique propre à générer une représentation graphique des échanges entre les constituants du système.

Chaque échange est caractérisé par un constituant émetteur, un constituant récepteur, et l'information transportée lors de cet échange.

La Figure 2 illustre un exemple de représentation graphique de trois modèles MSC élémentaires, notés MSC_1, MSC_2 et MSC_3, associés à trois composants C_1, C_2 et C_3 respectivement d'une chaîne fonctionnelle.

Chaque représentation graphique comprend un rectangle 30, représentant le constituant C (C_1, C_2 ou C_3) auquel est associé le modèle MSC élémentaire, et une ligne verticale 32 formant une ligne de temps. Chaque représentation graphique illustre en outre la séquence d'échanges mise en œuvre par le constituant associé, sous la forme de flèches incidentes 33a sur la ligne de temps, correspondant à des messages reçus par le constituant, ou de flèches 33b issues de la ligne de temps, correspondant à des messages émis par le constituant. La position de ces flèches par rapport à la ligne de temps est représentative de l'ordonnancement des messages émis ou reçus par le constituant.

Le modèle élémentaire MSC_1 illustré sur la Figure 2 représente ainsi un message m1 émis par le constituant C_1, et un message m3 reçu par la suite par le constituant C_1.

De même, le modèle élémentaire MSC_2 représente un message m1 reçu par le constituant C_2, et un message m2 émis par la suite par le constituant C_2.

Enfin, le modèle élémentaire MSC_3 représente un message m2 reçu par le constituant C_3, et un message m3 émis par la suite par le constituant C_3.

La représentation graphique illustre par ailleurs chaque contrainte temporelle associée à la séquence d'échanges, sous la forme d'une indication textuelle 34 de l'intervalle de temps associé à chaque contrainte, en relation avec les messages soumis à cette contrainte, et d'un symbole 36 représentatif du type de la contrainte, parmi les quatre types définis ci-dessus.

Par exemple, le modèle élémentaire MSC_1 illustré sur la Figure 2 représente une contrainte temporelle associée à la séquence comprenant l'émission du message m1 et à la réception du message m3 sous la forme d'une indication textuelle 34 indiquant que l'intervalle de temps associé à la contrainte est l'intervalle [0 ms, 30 ms] et d'un symbole

36 indiquant que la contrainte est du troisième type, c'est-à-dire une contrainte d'attente. Cette contrainte signifie donc qu'après émission du message m1, le composant C_1 est susceptible de prendre en compte un message m3 dans un délai compris entre 0 ms et 30 ms.

Chacun des modèles élémentaires MSC_2 et MSC_3 représentent une contrainte causale selon laquelle après réception du message m1, respectivement m2, le constituant C2, respectivement C3 émettra un message m2, respectivement m3 dans un délai compris entre 10 et 20 ms, respectivement entre 10 et 30 ms.

L'élément logiciel 26 est en outre propre à générer, par composition des modèles élémentaires MSC associés aux constituants de la chaîne fonctionnelle, un modèle global représentatif de l'ensemble du scénario, c'est-à-dire de l'ensemble des messages échangés lors de la réalisation du scénario. Ce modèle global est par exemple généré au moyen d'un élément HMSC, pour « High level MSC », de la norme MSC.

Le modèle global est par exemple généré de manière récursive, en combinant dans une première étape plusieurs modèles élémentaires pour former un ou des modèles composés, puis en combinant lors d'une ou plusieurs étapes successives un ou des modèles composés et/ou un/ou des modèles élémentaires jusqu'à obtention du modèle global. Notamment, deux modèles HMSC ou un modèle HMSC et un modèle MSC peuvent être combinés l'un à l'autre, et reliés par une contrainte temporelle.

La composition des modèles MSC et/ou HMSC est réalisée suivant une algèbre de composition comprenant trois opérateurs.

Un premier opérateur, noté « ET », commutatif et associatif, permet la composition en parallèle de modèles élémentaires MSC et/ou HMSC.

Un deuxième opérateur, noté « NEXT », associatif mais non commutatif, est destiné à la composition séquentielle de modèles MSC élémentaires et/ou HMSC. L'utilisation de cet opérateur permet par exemple de réutiliser des séquences intervenant plusieurs fois dans un même scénario.

Un troisième opérateur, noté « XOR », commutatif mais non associatif, permet d'exprimer une alternative entre deux modèles MSC élémentaires et/ou HMSC, ce qui permet notamment de modéliser des comportements nominaux ou dégradés du système.

Ce modèle MSC global est généré en identifiant les messages correspondants des différents modèles MSC élémentaires, ce qui permet de reconstituer l'ensemble des séquences d'échanges du scénario.

On a ainsi illustré sur la Figure 3 un modèle MSC global résultant de la composition, selon l'opérateur « ET », des modèles MSC_1, MSC_2, et MSC_3 de la Figure 2.

L'élément logiciel 28 est propre à déterminer si le modèle du scénario est valide ou non. En particulier, l'élément logiciel 28 est propre à vérifier que les contraintes temporelles du modèle MSC global qui sont définies sont compatibles les unes avec les autres, c'est-à-dire peuvent être toutes satisfaites.

Chaque type de contrainte est associé à un type de vérification.

En particulier, les contraintes causales et de synchronisation sont associées à un premier type de vérification. Selon ce premier type de vérification, une contrainte causale ou de synchronisation, associée à un intervalle de temps durant lequel un événement prédéterminé doit se produire, est considérée comme satisfaite si quel que soit l'instant, dans l'intervalle de temps, auquel l'évènement se produit, les autres contraintes de temps du modèle sont satisfaites.

Les contraintes des troisième et quatrième types sont associées à un deuxième type de vérification. Selon ce deuxième type de vérification, une contrainte de type attente ou exigence, associée à un intervalle de temps durant lequel un événement se produit, est considérée comme satisfaite s'il existe au moins un instant de l'intervalle de temps tel que si l'évènement se produit à cet instant, alors les autres contraintes de temps sont satisfaites.

Ainsi, l'élément logiciel 28 est propre à déterminer si toutes les combinaisons de valeurs dans les intervalles de temps associés aux contraintes du premier et du deuxième type, et au moins une valeur de l'intervalle de temps associé à chaque contrainte du troisième et du quatrième type permettent la mise en œuvre du scénario associé au modèle.

Si le modèle est sous-contraint, c'est-à-dire si la séquence ne peut être garantie par les contraintes exprimées, l'élément logiciel 28 est propre à vérifier, au sein de chaque chaîne temporelle continue de ce modèle, c'est-à-dire chaque suite d'échanges associés à des contraintes temporelles définies, que ces contraintes temporelles définies sont compatibles les unes avec les autres.

Deux cas de figure sont envisagés.

Un premier cas est celui d'une contrainte temporelle non définie associée à un premier et un deuxième message m1 et m2 tels que les échanges précédant le premier message m1 et ceux suivant le deuxième message m2 ne partagent aucune autre contrainte temporelle que la contrainte temporelle non défini.

Dans ce premier cas, l'élément logiciel 28, pour vérifier la validité du modèle, est propre à vérifier que les contraintes temporelles liées aux échanges précédant le premier message m1 sont compatibles entre elles, et que les contraintes temporelles liées aux échanges suivant le deuxième message m2 sont compatibles entre elles.

Un deuxième cas est celui d'une contrainte temporelle non définie associée à un premier et un deuxième messages m1 et m2, telle qu'il existe au moins une autre contrainte temporelle englobant la contrainte temporelle non définie, c'est-à-dire associée à deux message m1' et m2' tels que m1' correspond à m1 ou est antérieur à m1, et m2' correspond à m2 ou est postérieur à m2.

Dans ce deuxième cas, l'élément logiciel 28, pour vérifier la validité du modèle, est propre à vérifier que les contraintes temporelles liées aux échanges précédant m1 sont compatibles entre elles, et que les contraintes temporelles liées aux échanges suivant m2' sont compatibles entre elles.

Pour effectuer ces vérifications, l'élément logiciel 28 comprend un module 40 apte à générer, à partir des modèles MSC global et élémentaires, un modèle à automates temporisés du scénario, et un module 42 apte à exécuter ce modèle à automates temporisés pour vérifier la validité du modèle.

Le modèle à automates temporisés comprend, pour chacun des modèles MSC élémentaires, au moins un automate temporisé représentatif de la séquence d'échanges destinée à être mise en œuvre par le constituant associé au modèle MSC élémentaire, et de la ou des contrainte(s) temporelle(s) associée(s) à cette séquence d'échanges.

Ainsi, le module 40 est apte à générer, à partir de chaque modèle MSC élémentaire, au moins un automate à états finis représentatif de la séquence d'échanges destinée à être mise en œuvre par le constituant associé au modèle MSC élémentaire et de la ou des contrainte(s) temporelle(s) associée(s) à cette séquence d'échanges, et à générer un automate de synchronisation de ces automates à états finis, propre à synchroniser l'exécution des automates à états finis.

Le modèle à automates temporisés est par exemple implémenté au moyen de l'outil UPPAAL.

A cette fin, le module 40 est apte à générer un automate, noté Sc_n, pour chaque constituant de la chaîne fonctionnelle, à instancier, pour chaque contrainte temporelle définie de la séquence d'échanges mise en œuvre par ce constituant, un automate de contrainte, noté Ct_m, et à générer un automate de synchronisation AS.

Un automate Sc_n généré pour un constituant C_n traduit la séquence d'émission/réception de messages aux bornes de ce constituant par une séquence d'état dans un automate. La transition entre 2 états correspond à une émission ou à une réception de message, l'état correspond à une attente de cette réception/émission.

Chaque automate de contrainte Ct_m modélise l'attente dans un état de l'automate Sc_n, la durée de cette attente étant définie par l'intervalle de temps associée à la contrainte temporelle.

Chaque automate de contrainte Ct_m est ainsi initialisé lors de la réception ou de l'émission du premier message par le constituant et arrêté lors de la réception ou de l'émission du deuxième message par le constituant.

On a représenté sur les Figures 4 et 5, à titre d'exemple, deux automates Sc_1 et Sc_3 générés respectivement pour le constituant C_1 et le constituant C_3 de la Figure 2.

L'automate Sc_1 fait apparaître trois états successifs du constituant C_1, notés L_1, L_2 et L_3 sur la Figure 4, et deux transitions entre ces états, correspondant respectivement à l'émission du message m1 (notée m1!) et à la réception du message m3 (notée m3?) par le constituant C_1. Un automate de contrainte, correspondant à la contrainte d'attente entre l'émission de m1 et la réception de m3, est initialisé lorsque l'automate Sc_1 est dans l'état L_2, la transition entre L_2 et L_3 permettant l'arrêt de cet automate de contrainte.

L'automate Sc_3 fait de même apparaître trois états successifs du constituant C_3, notés L_1, L_2 et L_3 sur la Figure 5, et deux transitions entre ces états, correspondant respectivement à la réception du message m2 (notée m2?) et à l'émission du message m3 (notée m3!) par le constituant C_3. Un automate de contrainte, correspondant à la contrainte causale entre la réception de m2 et l'émission de m3, est initialisé lorsque l'automate Sc_3 est dans l'état L_2, la transition entre L_2 et L_3 se faisant à l'issue de l'exécution de l'automate de contrainte, et sous réserve que ce message puisse déjà être pris en considération par le constituant C_1.

Chaque automate de contrainte Ct_m comprend plusieurs états successifs, la transition entre ces états étant pilotée notamment par l'automate de synchronisation et en fonction de la valeur de variables booléennes.

Ces états comprennent notamment un état « initialisé », un état « démarré » et un état « arrêté ».

L'état « initialisé » est l'état de l'automate de contrainte lorsqu'il est initialisé par l'automate du constituant associé. La transition de l'état « initialisé » à l'état « démarré » est pilotée par l'automate de synchronisation. Enfin, la transition vers l'état « arrêté » n'est possible que si la contrainte de temps est satisfaite, ou si une discontinuité de la chaîne temporelle a été constatée, et est également pilotée par l'automate de synchronisation.

Les variables booléennes incluent, pour chaque automate de contrainte Ct_m, une variable BEGIN[m], propre à signaler le fait que l'automate de contrainte a été initialisé. La variable BEGIN[m] prend initialement la valeur « false ». Elle prend la valeur « true », lors de l'initialisation de l'automate de contrainte, et prend la valeur « false » une fois l'automate de contrainte démarré (c'est-à-dire lors de la transition entre l'état « initialisé » de l'automate de contrainte et son état « démarré », comme décrit ci-après).

Les variables booléennes incluent en outre, pour chaque automate de contrainte Ct_m, une variable END[m] qui permet de s'assurer que chaque automate est arrêté à l'issue d'une exécution donné des automates temporisés. La variable END[m] prend initialement la valeur « false», prend la valeur « true » lorsque l'automate Ct_m est prêt à être arrêté, c'est-à-dire lorsque le message mettant fin à la contrainte est reçu ou émis par le constituant, et reprend la valeur « false» lors de l'arrêt de l'automate de contrainte Ct_m.

Ainsi, la combinaison des valeurs des variables BEGIN et END d'un automate de contrainte définit si cet automate a été démarré et/ou arrêté.

Chaque automate de contrainte Ct_m est également associé à une valeur Mini[m] égale à la borne inférieure de l'intervalle de temps de la contrainte, et à une valeur Maxi[m] égale à la borne supérieure de cet intervalle de temps.

Les variables booléennes incluent par ailleurs une variable globale GlobalContinu, permettant de tenir compte des cas dans lesquels une contrainte temporelle associée à deux messages échangés par un constituant n'est pas définie, résultant en une discontinuité de la chaîne temporelle.

Les variables booléennes incluent également, pour chaque automate de contrainte, une variable Continu[m] dont la valeur est recopiée, lors de la transition de l'état « Initialisé » à l'état « Démarré » de cet automate, depuis GlobalContinu.

L'automate de synchronisation est propre à permettre le démarrage et l'arrêt de chaque automate de contrainte, afin d'en synchroniser l'exécution.

On a illustré schématiquement sur la Figure 6 un automate de synchronisation AS pour la synchronisation d'automates de contraintes.

Cet automate comprend un état initial L_{AS1}, et deux états instantanés L_{AS2} et L_{As3}.

A partir de l'état initial L_{AS1}, s'il existe au moins un automate de contrainte dont la variable BEGIN[m] a la valeur « true » et si les variables END de tous les automates de contraintes sont à « false » (c'est-à-dire s'il existe au moins un automate Ct_m qui a été initialisé mais qui n'est pas encore démarré et qu'aucun automate n'est prêt à être arrêté), l'automate AS passe de l'état L_{AS1} à l'état L_{AS3}. Cette transition correspond notamment à la première initialisation d'un automate de contrainte lors de l'exécution du modèle. En effet, le premier automate à être initialisé doit pouvoir être démarré sans qu'un autre automate de contrainte soit arrêté. Lorsqu'une contrainte temporelle n'est pas définie, cette transition correspond également à l'initialisation du premier automate de contrainte de la chaîne temporelle suivant cette contrainte.

Dans l'état L_{AS3}, l'automate AS envoie à tous les automates Ct_m dans l'état « Initialisé » un message « Début ». L'émission de ce message est en effet nécessaire pour que des automates initialisés démarrent (c'est-à-dire passent à l'état « démarré »).

Alternativement, à partir de l'état initial L_{AS1}, s'il existe au moins un automate Ct_m dont la variable END[m] a la valeur « true » (c'est-à-dire s'il existe au moins un automate Ct_m qui est prêt à être arrêté), l'automate de synchronisation AS passe dans l'état L_{AS2} en envoyant à tous les automates Ct_m prêts à être arrêtés un message « Fin ». L'émission de ce message est en effet nécessaire pour que ces automates puissent être arrêtés.

Dès que tous les automates Ct_m prêts à être arrêtés sont effectivement arrêtés, l'automate de synchronisation passe dans l'état L_{AS3}.

Ces états L_{AS1}, L_{AS2} et L_{AS3} permettent de vérifier qu'à chaque fois qu'une contrainte démarre, une autre se termine, si la chaîne temporelle de l'ensemble du modèle est continue. Ces états L_{AS1}, L_{AS2} et L_{AS3} permettent également de prendre en compte les cas dans lesquels la chaîne temporelle est au contraire discontinue et comprend au moins deux chaînes temporelles élémentaires continues, en permettant, lors de l'exécution du modèle à automates temporisés, de modéliser chacune de ces chaînes élémentaires et d'en vérifier la validité.

En effet, à partir de l'état L_{AS3}, l'automate AS passe à l'état L_{AS1} en fixant la valeur de la variable GlobalContinu à « false ». Cette valeur sera par la suite fixée à nouveau à « true » si un automate de contrainte peut être arrêté, comme décrit ci-après. Si au contraire aucun automate de contrainte n'est prêt à être arrêté alors qu'un nouvel automate de contrainte a été initialisé (en raison d'une contrainte temporelle non définie), la valeur de la variable GlobalContinu reste à « false », et ce nouvel automate de contrainte sera exécuté en conséquence, comme décrit ci-après.

Les automates de contraintes Ct_m liés aux contraintes temporelles sont générés conformément à un modèle type, qui dépend du type de vérification auquel est soumise la contrainte.

On a illustré sur la Figure 7 un automate de contrainte Ct_m selon un premier type, associé à un automate généré par le module 40 pour les contraintes vérifiées selon le premier type de vérification, c'est-à-dire les contraintes causales et de synchronisation.

L'automate de contrainte Ct_m est destiné à simuler le comportement du constituant associé entre la réception ou l'émission d'un premier message, noté m_A, et l'émission d'un deuxième message, noté m_B, lorsque le message m_B est émis dans un délai égal à la borne inférieure de l'intervalle de temps de la contrainte, et lorsque le message m_B est émis dans un délai égal à la borne supérieure de cet intervalle de temps.

L'automate de contrainte Ct_m permet ainsi de vérifier si, que le message m_B soit émis ou reçu dans un délai égal à la borne inférieure ou supérieure de l'intervalle de temps, les autres contraintes temporelles peuvent également être satisfaites. Si c'est le cas, on peut en effet en déduire que quel que soit le moment de cet intervalle de temps

auquel le message m_B est émis ou reçu, les autres contraintes temporelles peuvent également être satisfaites.

L'automate de contrainte Ct_m comprend un état initialisé, noté LocStart, suivi d'un état démarré noté LocBegin.

Pour passer de l'état LocStart à l'état LocBegin, il est nécessaire que l'automate de contrainte Ct_m ait été initialisé par l'automate Sc du constituant associé par passage de la valeur de la variable BEGIN[m] à « true », ce qui nécessite que le message m_A ait été émis par le constituant associé. Il est également nécessaire que l'automate Ct_m reçoive un message « Début » de l'automate de synchronisation, c'est-à-dire que l'automate de synchronisation soit dans l'état L_{AS3} soit suite à l'arrêt d'une autre contrainte, soit du fait que l'automate Ct_m est le premier à être initialisé ou le premier à être initialisé après une rupture de la chaîne temporelle.

Lors de cette transition, l'automate de contrainte Ct_m initie une horloge CLOCK[m] à 0, fixe la valeur de la variable BEGIN[m] à « false », et recopie la valeur de la variable GlobalContinu dans une variable Continu[m].

L'état de démarrage LocBegin est un état à partir duquel l'automate doit instantanément passer dans un autre état.

Si la variable Continu[m] a la valeur « true », c'est-à-dire si la contrainte temporelle est reliée à une chaîne temporelle, il convient de vérifier si la contrainte temporelle est compatible avec les contraintes temporelles de cette chaîne.

L'automate Ct_m passe alors, de manière aléatoire, soit dans un état LocMin, dans lequel il reste au plus tard jusqu'à ce que l'horloge CLOCK soit égale à Mini[m] (c'est-à-dire la borne inférieure de l'intervalle de temps de la contrainte), soit dans un état LocMax, dans lequel il reste au plus tard jusqu'à ce que l'horloge CLOCK soit égale à Maxi[m] (c'est-à-dire la borne supérieure de l'intervalle de temps de la contrainte).

A partir de chacun des états LocMin et LocMax, l'automate Ct_m passe dans un état final LocEnd sous réserve que la variable END[m] ait la valeur « true », c'est-à-dire que le message m_B ait été émis, que l'automate Ct_m reçoive un message « Fin » de l'automate de synchronisation, et que l'horloge CLOCK soit égale à Mini[m] (respectivement Maxi[m]). Lors du passage de l'état LocMin (respectivement LocMax) à l'état LocEnd, la valeur de END[m] est fixée à « false » et la valeur de la variable GlobalContinu à « true ».

Si au contraire la variable Continu[m] a la valeur « false », l'automate Ct_m passe alors directement de l'état LocBegin à l'état final LocEnd. En effet, cette situation se produit lorsqu'une contrainte temporelle n'est pas définie, dans le deuxième cas de figure présenté ci-dessus, pour les contraintes temporelles associées aux messages échangés entre les messages m1 et m2'. Il n'est alors pas nécessaire de vérifier que la contrainte

temporelle est vérifiée. En effet, dans ce cas, la valeur de la variable GlobalContinu, fixée à « false » lors de la rupture de la chaîne temporelle, du fait d'un automate de contrainte a été démarré sans qu'un autre soit arrêté, reste à cette valeur « false » tant que la contrainte temporelle englobant la contrainte non définie n'est pas arrêtée.

On a par ailleurs illustré sur la Figure 8 un automate de contrainte $Ct_{m'}$ selon un deuxième type, généré par le module 40 pour les contraintes vérifiées selon le deuxième type de vérification, c'est-à-dire les contraintes de type « attente » et « exigence ».

L'automate de contrainte $Ct_{m'}$ est destiné à simuler l'attente du constituant associé entre la réception ou l'émission d'un premier message, noté $m_{A'}$ et la réception d'un deuxième message, noté $m_{B'}$, ce message $m_{B'}$ devant être reçu dans un intervalle [Mini[m'], Maxi[m']] à partir du message $m_{A'}$.

Cet automate diffère de l'automate de la figure 6 en ce que, à partir de l'état LocBegin, l'automate Ct_m passe soit dans un état LocMax, soit directement dans l'état LocEnd, en fonction de la valeur de la variable Continu[m'].

Si Continu[m'] a la valeur « true », l'automate passe dans l'état LocMax dans lequel il reste au plus tard jusqu'à ce que l'horloge CLOCK soit égale à Max[m'].

Pour que l'automate Ct_m' passe de l'état LocMax à l'état final LocEnd, il est nécessaire que la variable END[m'] ait la valeur « true », c'est-à-dire que le message $m_{B'}$ ait été reçu, que l'automate $Ct_{m'}$ reçoive un message « Fin » de l'automate de synchronisation, et que l'horloge CLOCK soit comprise entre Mini[m'] et Maxi[m']. Dès que ces conditions sont rempliées, l'automate $CT_{m'}$ passe de l'état LocMax à l'état LocEnd, en fixant la valeur de END[m] à « false » et en fixant la valeur de la variable GlobalContinu à « true ».

Si le message $m_{B'}$ est reçu par le constituant après le délai Maxi[m'], l'automate reste bloqué dans l'état LocMax. L'automate de contrainte associé à l'émission du message $m_{B'}$ reste également bloqué, ce qui permet d'identifier les deux contraintes correspondantes comme incompatibles l'une avec l'autre.

Le module 42 est apte à déterminer si le modèle spécifique au scénario est valide, à partir du modèle à automates temporisés correspondant. Le module 42 est apte à considérer le modèle comme valide si toutes les contraintes temporelles sont compatibles entre elles, ou comme non valide si au moins deux contraintes temporelles sont incompatibles entre elles.

Pour cela, le module 42 est apte à vérifier, à partir du modèle à automates temporisés, si les contraintes temporelles associées aux échanges du scénario sont compatibles entre elles.

A cette fin, le module 42 est apte à exécuter le modèle à automates temporisés pour simuler la mise en œuvre du scénario, et à vérifier que cette simulation peut être menée à son terme, tous les automates de contrainte ayant été démarrés et arrêtés une fois.

En particulier, afin de vérifier que toutes les combinaisons de valeurs dans les intervalles de temps associés aux contraintes du premier et du deuxième type et au moins une valeur de l'intervalle de temps associé à chaque contrainte du troisième et du quatrième type permettent de mener la simulation à son terme, le module 42 garantit que quelque soit la combinaison d'états LocMin et LocMax pour toutes les contraintes causales, la fin du scénario est accessible. Cette vérification peut être réalisée de manière récursive.

Si le modèle est sous contraint, c'est-à-dire s'il existe une contrainte de temps non définie, le module 42 est propre à vérifier que l'exécution de la ou des chaîne(s) temporelle(s) excluant cette contrainte peut être menée à son terme.

Le modèle du scénario est jugé non valide par le module 42 s'il existe au moins une combinaison d'états LocMin ou LocMax pris par les automates associés aux contraintes causales et de synchronisation lors de l'exécution du modèle à automates temporisés telle que cette exécution n'est pas complète, par exemple parce qu'au moins un automate associé à une contrainte de type attente ou exigence reste bloqué dans l'état LocMax. En effet, une telle situation traduit une incompatibilité entre la contrainte associée à l'automate bloqué et les autres contraintes.

Le modèle du scénario est jugé valide par le module 42, ou valide sous condition, si au moins une contrainte n'est pas spécifiée, si quelle que soit la combinaisons d'états LocMin ou LocMax prise par les automates associés aux contraintes causales et de synchronisation lors de l'exécution du modèle à automates temporisés, l'exécution de ce modèle, ou, dans le cas d'un modèle sous contraint, l'exécution des chaînes temporelles excluant la contrainte non définie, est complet, et ne conduit à aucun blocage.

Le module 44 de restitution est apte à recevoir du module 42 des informations relatives à la validité du modèle, indiquant notamment si le modèle est valide ou valide sous condition, ou, si le modèle est invalide, quelle(s) contrainte(s) est/sont incompatible(s) avec les autres contraintes.

En outre, le module 44 de restitution est propre à commander l'affichage, sur l'écran d'affichage 16, d'une représentation graphique du modèle MSC global du scénario, et d'indications graphiques indiquant si le modèle est valide, valide sous condition, ou non valide, et signalant, le cas échéant, quelles contraintes temporelles sont incompatibles entre elles. Cette signalisation est par exemple réalisée sous la forme d'un

code couleur. Par exemple, l'indication textuelle 34 de l'intervalle de temps associé à une contrainte est surligné en vert si cette contrainte est compatible avec toutes les autres, en orange si cette contrainte n'a pas pu être vérifiée, et en rouge si cette contrainte est incompatible avec au moins une autre contrainte (celle-ci étant alors également signalée par un surlignage rouge).

On a illustré sur la Figure 9 un exemple d'une telle représentation graphique, représentant le modèle MSC global de la Figure 3, sur laquelle figure en outre une indication 50 selon laquelle le modèle n'est pas valide, et sur laquelle les contraintes temporelles non compatibles entre elles sont signalées, dans le cas présent sous la forme d'un surlignage 52, notamment de couleur rouge. Cette signalisation indique que la contrainte temporelle, de type exigence, liée à l'émission du message m1 et à la réception du message m3 par le constituant C_1, n'est pas compatible avec la contrainte temporelle, de type causale, liée à la réception du message m2 et à l'émission du message m3 par le constituant C_3. En effet, il existe des valeurs de l'intervalle de temps associé à cette dernière contrainte pour lesquelles la contrainte temporelle associée à C_1 ne peut pas être satisfaite.

Le module 44 de restitution est également apte à commander l'affichage, sur l'écran d'affichage 16, d'une représentation graphique des modèles MSC élémentaires associés aux contraintes incompatibles entre elles.

Un exemple de mise en œuvre d'un procédé de détermination de la validité d'un modèle représentatif d'un scénario d'utilisation d'une chaîne fonctionnelle d'un système complexe selon un mode de réalisation, au moyen du système de la Figure 1, va maintenant être décrit en référence à la Figure 10. Ce scénario comprend la mise en œuvre, par chacun d'une pluralité de constituants de la chaîne fonctionnelle, d'une séquence d'échanges de messages avec au moins un autre constituant de la chaîne fonctionnelle. Chaque séquence d'échanges est soumise à au moins une contrainte temporelle associée, qui peut être définie ou non définie.

Le procédé comprend une étape 102 de fourniture, par l'élément logiciel 24, d'une architecture hiérarchisée du système complexe, incluant les constituants, leurs interfaces, les fonctions associées et les liens entre les constituants.

Puis, le procédé comprend une étape 104 de génération, à l'aide de l'élément logiciel 26 d'un modèle MSC global spécifique à un scénario d'utilisation du système complexe, sur la base de l'architecture du système complexe fournie par l'élément logiciel 24. Ceci permet de s'assurer que l'architecture dynamique liée au scénario est cohérente avec l'architecture statique du système complexe.

Cette étape 104 comprend une phase 106 d'écriture d'un modèle MSC élémentaire pour chaque constituant de la chaîne fonctionnelle, par exemple en fonction de données relatives au scénario modélisé stockées dans la mémoire 22 ou saisies par un opérateur via l'interface homme machine 18.

Comme décrit ci-dessus et illustré sur la Figure 3, chaque modèle MSC élémentaire est représentatif de la séquence d'échanges de messages par le constituant avec d'autres constituants lors de la mise en œuvre du scénario, et des contraintes temporelles associées à cette dite séquence d'échanges.

Lors de cette étape, au moins une partie des contraintes temporelles est spécifiée.

L'étape 104 comprend par ailleurs une phase 108 d'écriture d'un modèle HMSC représentatif de l'ensemble du scénario, c'est-à-dire de l'ensemble des messages échangés lors de la réalisation du scénario, par composition des modèles MSC élémentaires associés aux constituants de la chaîne fonctionnelle. Le mode de composition des modèles MSC élémentaires, c'est-à-dire les opérateurs utilisés pour cette composition, est par exemple défini par un opérateur via l'interface homme machine 18. Lors de la phase 106, l'élément logiciel 26 génère ainsi le modèle MSC global à partir du mode de composition défini par l'opérateur. Comme décrit ci-dessus, le modèle HMSC global est par exemple généré de manière récursive.

Puis, le procédé comprend une étape 110 de génération, par le module 40, d'un modèle à automates temporisés du scénario, à partir des modèles MSC global et élémentaires.

Le modèle à automates temporisés comprend, pour chacun des modèles MSC élémentaires, un automate temporisé représentatif de la séquence d'échanges destinée à être mise en œuvre par le constituant associé au modèle MSC élémentaire, et de la ou des contrainte(s) temporelle(s) associée(s) à cette séquence d'échanges.

Ainsi, lors de l'étape 110, le module 40 génère, à partir de chaque modèle MSC élémentaire, un automate à états finis représentatif de la séquence d'échanges destinée à être mise en œuvre par le constituant associé au modèle MSC élémentaire et de la ou des contrainte(s) temporelle(s) associée(s) à cette séquence d'échanges.

Le module 40 génère également un automate de synchronisation de ces automates à états finis, propre à synchroniser l'exécution des automates à états finis.

La génération d'un automate temporisé associé à un modèle MSC élémentaire comprend la génération d'un automate à états finis Sc représentatif de la séquence d'échanges destinée à être mise en œuvre par le constituant associé, et la génération, pour chacune des contraintes temporelles associées à la séquence d'échanges, d'un automate de contrainte Ct représentatif de cette contrainte temporelle.

Lors de l'étape 110, les automates de contraintes Ct liés aux contraintes temporelles sont générés conformément à un modèle type, qui dépend du type de vérification auquel est soumise la contrainte, comme décrit précédemment. L'automate de synchronisation est par exemple conforme à l'automate illustré sur la Figure 5. Les automates Sc associés aux constituants sont en revanche chacun représentatif de la séquence d'échange mise en œuvre par ce constituant.

Puis, lors d'une étape 112, l'élément logiciel 28 détermine si le modèle représentatif du scénario est valide ou non, à partir du modèle à automates temporisés généré lors de l'étape 110. modèles MSC élémentaires et du modèle MSC global modélisant ce scénario. Lors de l'étape 112, l'élément logiciel 28 vérifie ainsi que les contraintes temporelles du modèle sont compatibles les unes avec les autres.

Lors de l'étape 112, le module 42 vérifie, à partir des automates temporisés, si les contraintes temporelles associées aux séquences d'échanges du scénario sont compatibles entre elles.

A cette fin, le module 42 réalise un calcul d'accessibilité de l'état final du scénario. Cette vérification peut être réalisée de manière récursive.

Si le modèle est sous contraint, c'est-à-dire s'il existe une contrainte de temps non définie, le module 42 vérifie que l'exécution de la ou des chaîne(s) temporelle(s) excluant cette contrainte peut être menée à son terme, tous les automates de contrainte de cette chaîne temporelle ayant été démarrés et arrêtés au moins une fois.

Le modèle du scénario est jugé non valide par le module 42 s'il existe au moins une combinaison d'états LocMin ou LocMax pris par les automates associés aux contraintes causales et de synchronisation lors de l'exécution du modèle à automates temporisés telle que cette exécution n'est pas complète, par exemple parce qu'au moins un automate de contrainte associé à une contrainte de type attente ou exigence reste bloqué dans l'état LocMax. En effet, une telle situation traduit une incompatibilité entre la contrainte associée à l'automate bloqué et les autres contraintes.

Le modèle du scénario est jugé valide par le module 42, ou valide sous condition, si au moins une contrainte n'est pas spécifiée, si quelle que soit la combinaison d'états LocMin ou LocMax prise par les automates associés aux contraintes causales et de synchronisation lors de l'exécution du modèle à automates temporisés, l'exécution du modèle, ou, dans le cas d'un modèle sous contraint, l'exécution des chaînes temporelles excluant la contrainte non définie, est complet, et ne conduit à aucun blocage.

Puis, lors d'une étape 120, le module 44 de restitution reçoit du module 42 des informations relatives à la validité du modèle, indiquant notamment si le modèle est valide

ou valide sous condition, ou, si le modèle est invalide, quelle(s) contrainte(s) est/sont incompatible(s) avec les autres contraintes.

En outre, le module 44 de restitution commande l'affichage, sur l'écran d'affichage 16, d'une représentation graphique du modèle MSC global du scénario, et d'indications graphiques indiquant si le modèle est valide, valide sous condition, ou non valide, et signalant, le cas échéant, quelles contraintes temporelles sont incompatibles entre elles.

Le module 44 de restitution commande également l'affichage sur l'écran d'affichage 16, par exemple en réponse à une action d'un opérateur sur l'interface homme-machine 18, d'une représentation graphique des modèles MSC élémentaires associés aux contraintes incompatibles entre elles.

Le système et le procédé selon l'invention permettent ainsi de vérifier formellement la validité d'un modèle d'un scénario, entièrement contraint ou sous-contraint, tout en garantissant la cohérence entre l'architecture hiérarchisée du système et les séquences d'échanges destinées à être mises en œuvre par ce système.

En particulier, la génération du modèle à automate temporisés à partir d'un modèle MSC du scénario, lui-même généré conformément à l'architecture du système, permet d'assurer la cohérence entre le modèle statique du système et le modèles dynamique du scénario.

Par ailleurs, l'utilisation du modèle MSC pour modéliser le scénario permet aux opérateurs d'appréhender de façon simple et claire les séquences d'échanges entre les constituants, tandis que l'utilisation d'un modèle à automates temporisés permet de vérifier de façon formelle que les contraintes temporelles du modèle sont compatibles entre elles.

En outre, le modèle à automates temporisés est généré de façon rapide et a une grande lisibilité. En effet, lors de la génération de ce modèle, seuls les automates associés aux constituants sont spécifiques au scénario modélisé, les autres automates étant conformes à des modèles prédéfinis. Par ailleurs, la modélisation du scénario au moyen de plusieurs automates chacun associé à un constituant ou une contrainte en facilite la lecture et l'interprétation.

Il devra être compris que les exemples de réalisation présentés ci-dessus ne sont pas limitatifs.

Notamment, selon un autre mode de réalisation, le modèle à automates temporisés est généré au moyen d'un autre outil que l'outil UPPAAL, par exemple en traduisant les modèles MSC élémentaires en language Promela (Process Meta Language) et en vérifiant la compatibilité des contraintes au moyen de l'outil SPIN (« Simple Promela Interpreter »), ou au moyen de l'outil LTSA.

REVENDICATIONS

1.- Procédé mis en œuvre par ordinateur de détermination d'une validité d'un modèle représentatif d'un scénario d'utilisation d'une chaîne fonctionnelle d'un système complexe, ledit scénario comprenant la mise en œuvre, par au moins un constituant de la chaîne fonctionnelle, d'une séquence d'échanges de messages avec au moins un autre constituant de ladite chaîne fonctionnelle ou un élément extérieur audit système complexe, au moins une séquence d'échanges étant soumise à au moins une contrainte temporelle associée,

ledit procédé étant caractérisé en ce qu'il comprend :

- une étape (104) de fourniture d'un modèle représentatif de l'ensemble des séquences d'échanges destinés à être mise en œuvre par le ou lesdits constituants (C) lors de la mise en œuvre dudit scénario,

- une étape (110) de génération, à partir dudit modèle, d'un modèle à automates temporisés dudit scénario,

- une étape (112) de vérification, à partir dudit modèle à automates temporisés, d'une compatibilité entre les contraintes temporelles associées aux séquences d'échanges dudit scénario, ledit modèle étant considéré comme non valide si au moins deux contraintes temporelles sont incompatibles entre elles.

2.- Procédé selon la revendication 1, caractérisé en ce que l'étape (104) de fourniture dudit modèle comprend :

- une phase (106) de génération, pour chaque constituant de ladite chaine fonctionnelle, d'un modèle élémentaire représentatif de la séquence d'échanges destinée à être mise en œuvre par ledit constituant et de la ou des contrainte(s) temporelle(s) associée(s) à ladite séquence d'échanges,

- une phase (108) de détermination dudit modèle représentatif dudit scénario, par composition desdits modèles élémentaires.

3.- Procédé selon la revendication 2, caractérisé en ce que l'étape (110) de génération dudit modèle à automates temporisés comprend la génération, à partir de chacun desdits modèles élémentaires, d'au moins un automate temporisé (Sc, Ct) représentatif de la séquence d'échanges destinée à être mise en œuvre par ledit constituant et de la ou des contrainte(s) temporelle(s) associée(s) à ladite séquence d'échanges, et la génération d'un automate de synchronisation (AS) desdits automates temporisés (Sc, Ct).

4.- Procédé selon la revendication 3, caractérisé en ce que la génération de chaque automate temporisé (Sc, Ct) comprend la génération d'un automate à états finis

(Sc) représentatif de la séquence d'échanges destinée à être mise en œuvre par ledit constituant (C), et la génération, pour chacune des contraintes temporelles associées à ladite séquence d'échanges, d'un automate de contrainte (Ct) représentatif de ladite contrainte temporelle.

5.- Procédé selon l'une quelconque des revendications 2 à 4, caractérisé en ce que chaque modèle élémentaire est un modèle élémentaire de type MSC, et en ce que ladite phase (108) de détermination dudit modèle représentatif du scénario comprend une composition desdits modèles élémentaires de type MSC, ledit modèle représentatif du scénario étant de type HMSC.

6.- Procédé selon la revendication 5, caractérisé en ce que la composition desdits modèles MSC élémentaires est réalisée suivant une algèbre de composition comprenant :

- un premier opérateur (ET), commutatif et associatif,
- un deuxième opérateur (NEXT) associatif et non commutatif,
- un troisième opérateur (XOR) commutatif et non associatif.

7.- Procédé selon l'une quelconque des revendications précédentes, caractérisé en ce que chaque contrainte temporelle est associée à un intervalle de temps, à partir d'un événement de référence, dans lequel un événement prédéfini est attendu.

8.- Procédé selon la revendication 7, caractérisé en ce que chaque contrainte temporelle est associée à un type de vérification choisi parmi :

- un premier type de vérification, tel que la contrainte temporelle est considérée comme satisfaite si, quel que soit l'instant de l'intervalle de temps auquel ledit événement se produit, les autres contraintes temporelles peuvent également être satisfaites; et
- un deuxième type de vérification, tel que la contrainte temporelle est considérée comme satisfaite s'il existe au moins un instant de l'intervalle de temps auquel l'événement se produit, permettant aux autres contraintes temporelles d'être satisfaites.

9.- Procédé selon la revendication 8, caractérisé en ce que lors de ladite étape (112) de vérification, les contraintes temporelles sont jugées compatibles entre elles si lors de la mise en œuvre du scénario, toutes les contraintes temporelles peuvent être satisfaites.

10.- Procédé selon l'une quelconque des revendication 7 à 9, caractérisé en ce que ledit procédé comprend la spécification d'au moins une partie des contraintes temporelles, la spécification d'une contrainte temporelle comprenant la définition de la valeur des bornes inférieure et supérieure de l'intervalle de temps associé à ladite contrainte temporelle.

11.- Procédé selon la revendication 10, caractérisé en ce qu'au moins une contrainte temporelle n'est pas spécifiée, et en ce que lors de l'étape de vérification, le

scénario est considéré comme valide, sous réserve de ladite contrainte temporelle non spécifiée, si toutes les contraintes temporelles spécifiées sont compatibles entre elles.

12.- Procédé selon la revendication 10, caractérisé en ce que toutes les contraintes temporelles sont spécifiées, et en ce que lors de l'étape (112) de vérification, le scénario est considéré comme valide si toutes les contraintes temporelles sont compatibles entre elles.

13.- Procédé selon l'une quelconque des revendications 7 à 12, caractérisé en ce que chacun desdits événement de référence et événement prédéfini est une réception ou une émission d'un message par le constituant destiné à mettre en œuvre la séquence d'échanges soumise à ladite contrainte temporelle.

14.- Procédé selon l'une quelconque des revendications 7 à 13, caractérisé en ce que chaque contrainte temporelle est d'un type choisi parmi :

- un premier type de contraintes selon lequel ledit événement de référence et ledit événement prédéfini sont causalement dépendant ;
- un deuxième type de contraintes destiné à assurer une synchronisation temporelle entre deux constituants ;
- un troisième type de contraintes destiné à spécifier un intervalle de temps, après ledit événement de référence, dans lequel un constituant est susceptible de prendre en compte un message qu'il reçoit d'un autre constituant ou d'un élément extérieur au système complexe ;
- un quatrième type de contraintes destiné à spécifier une exigence temporelle garantissant un bon fonctionnement du système.

15.- Système (10) de détermination d'une validité d'un modèle représentatif d'un scénario d'utilisation d'une chaîne fonctionnelle d'un système complexe, ledit scénario comprenant la mise en œuvre, par au moins un constituant de la chaîne fonctionnelle, d'une séquence d'échanges de messages avec au moins un autre constituant de ladite chaîne fonctionnelle ou un élément extérieur audit système complexe, chaque séquence d'échanges étant soumise à au moins une contrainte temporelle associée,

ledit système étant caractérisé en ce qu'il comprend :

- des moyens (26) de fourniture d'un modèle représentatif de l'ensemble des séquences d'échanges destinés à être mise en œuvre par le ou lesdits constituants (C) lors de la mise en œuvre dudit scénario,
- un élément logiciel (28) de vérification de la validité dudit modèle, ledit élément logiciel (28) de vérification comprenant :
  - un module (40) pour générer, à partir dudit modèle, un modèle à automates temporisés dudit scénario,

- un module (42) pour vérifier, à partir dudit modèle à automates temporisés, une compatibilité entre les contraintes temporelles associées aux séquences d'échanges dudit scénario, ledit modèle étant considéré comme non valide si au moins deux contraintes temporelles sont incompatibles entre elles.

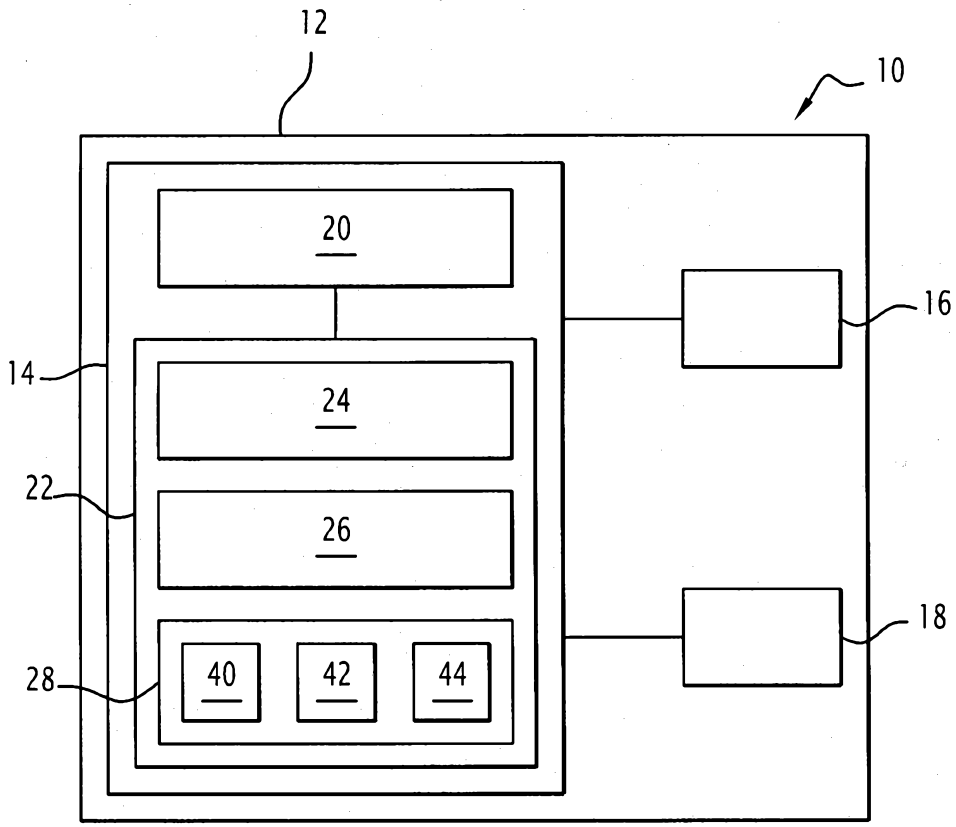

FIG.1

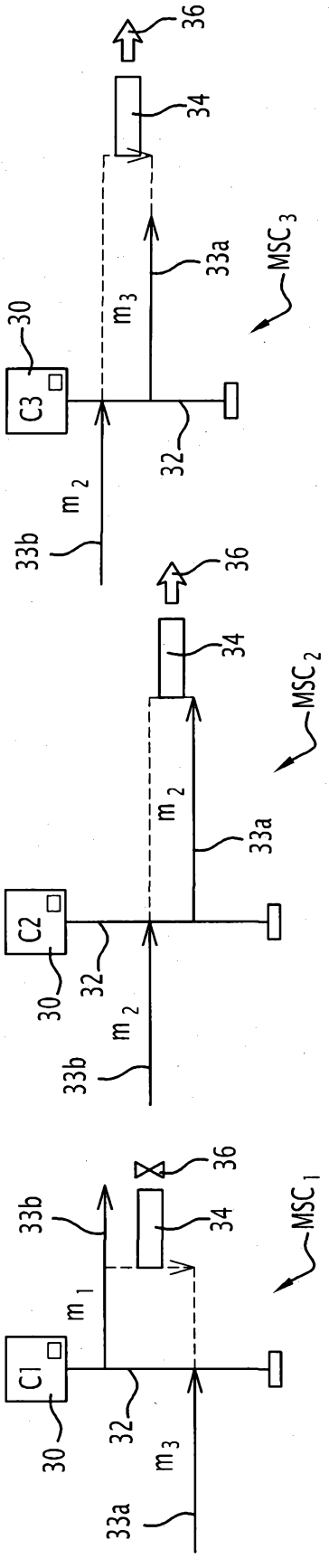
C1
C2
C3
30
32
33a
33b
34
36
m1
m2
m3
MSC1
MSC2
MSC3

FIG.2

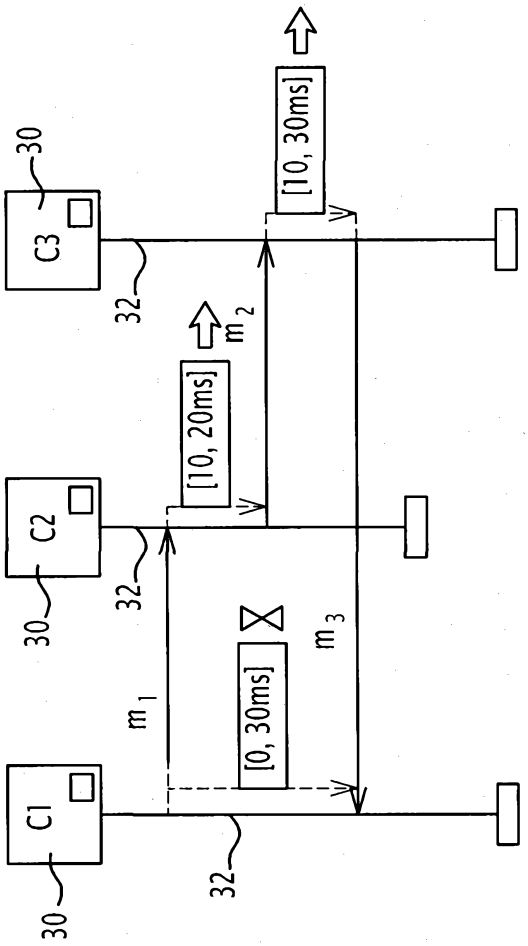
C1
C2
C3
30
32
m1
m2
m3
[0, 30ms]
[10, 20ms]
[10, 30ms]

FIG.3

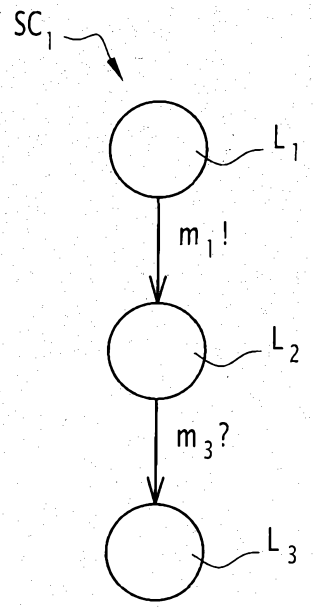

FIG.4

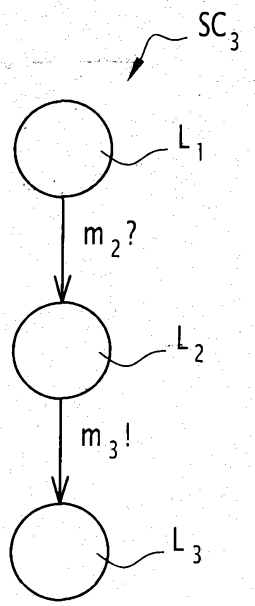

FIG.5

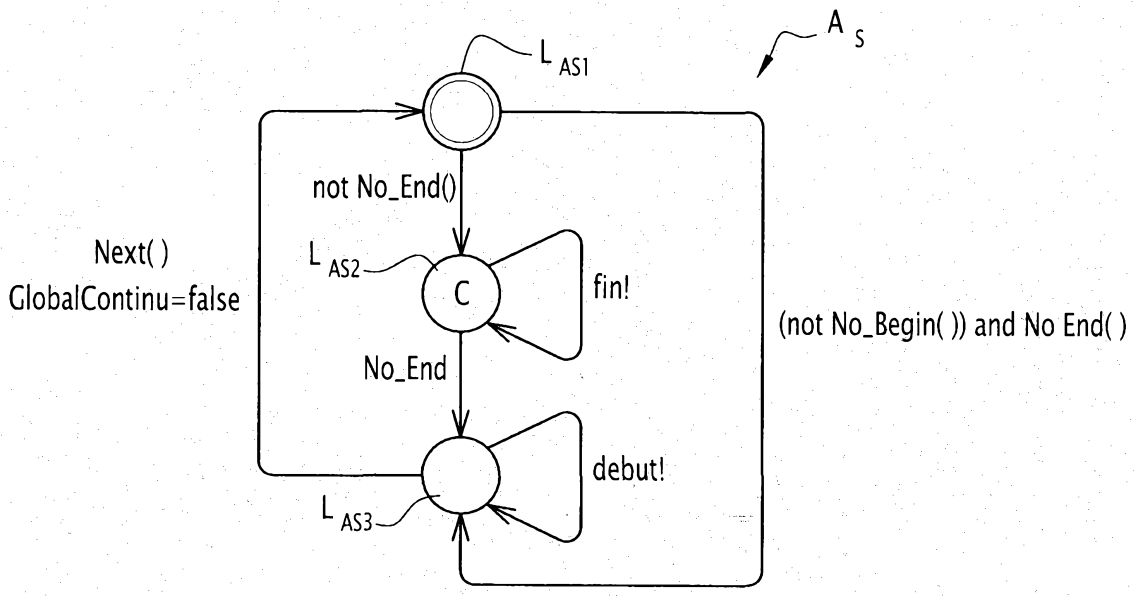

FIG.6

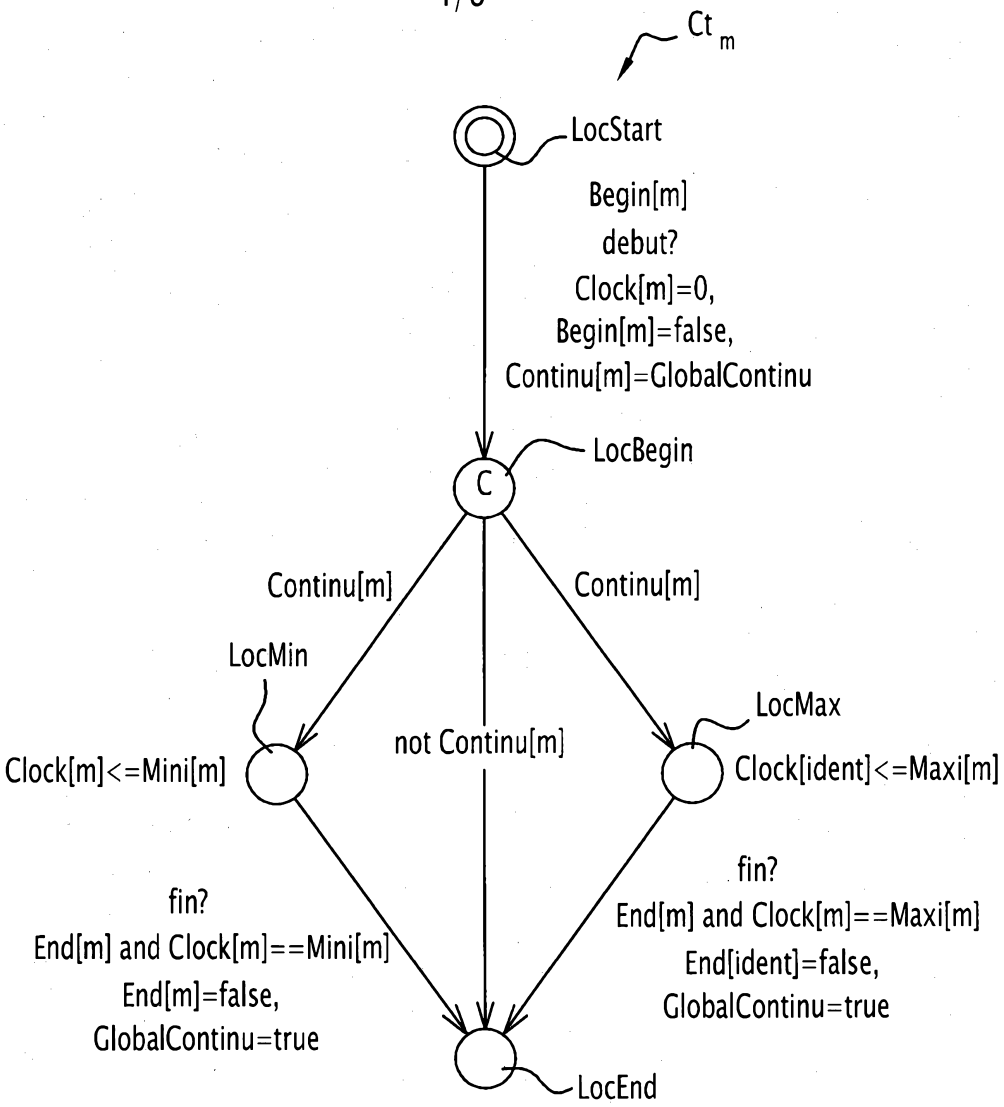
Ct m
LocStart
Begin[m]
debut?
Clock[m]=0,
Begin[m]=false,
Continu[m]=GlobalContinu
LocBegin
C
Continu[m]
Continu[m]
LocMin
LocMax
not Continu[m]
Clock[m]<=Mini[m]
Clock[ident]<=Maxi[m]
fin?
End[m] and Clock[m]==Mini[m]
End[m]=false,
GlobalContinu=true
fin?
End[m] and Clock[m]==Maxi[m]
End[ident]=false,
GlobalContinu=true
LocEnd

FIG.7

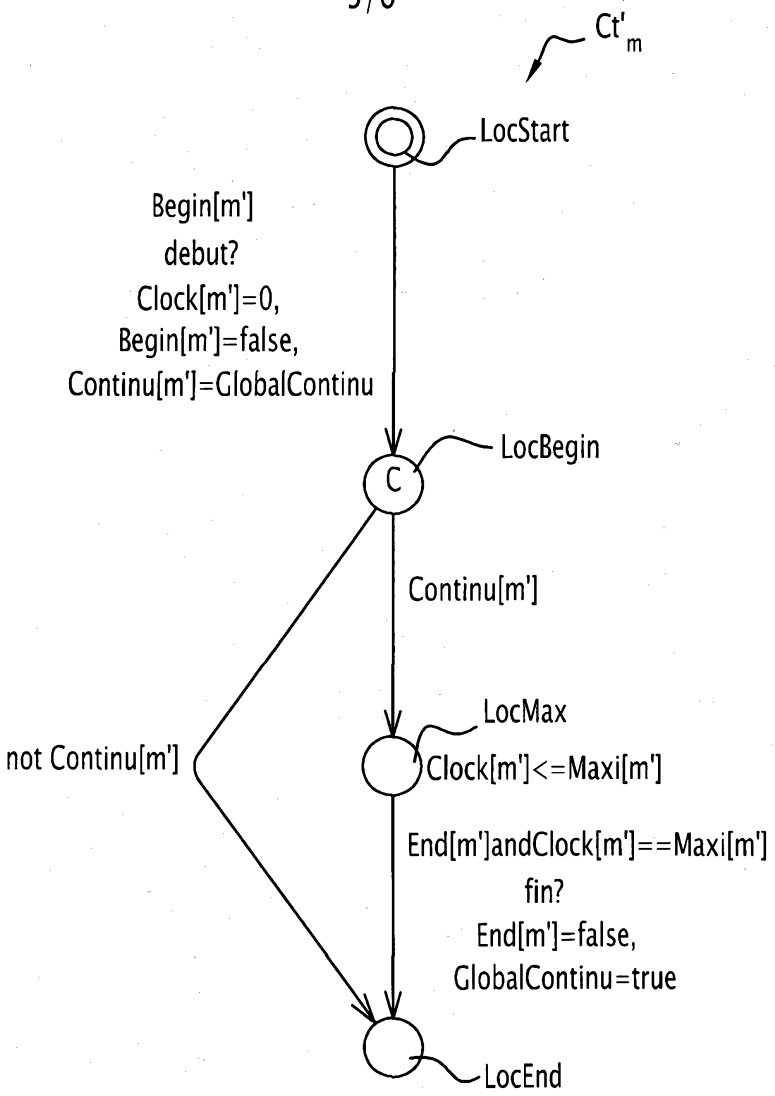
Ct'm
LocStart
Begin[m']
debut?
Clock[m']=0,
Begin[m']=false,
Continu[m']=GlobalContinu
LocBegin
C
Continu[m']
LocMax
not Continu[m']
Clock[m']<=Maxi[m']
End[m']andClock[m']==Maxi[m']
fin?
End[m']=false,
GlobalContinu=true
LocEnd

FIG.8

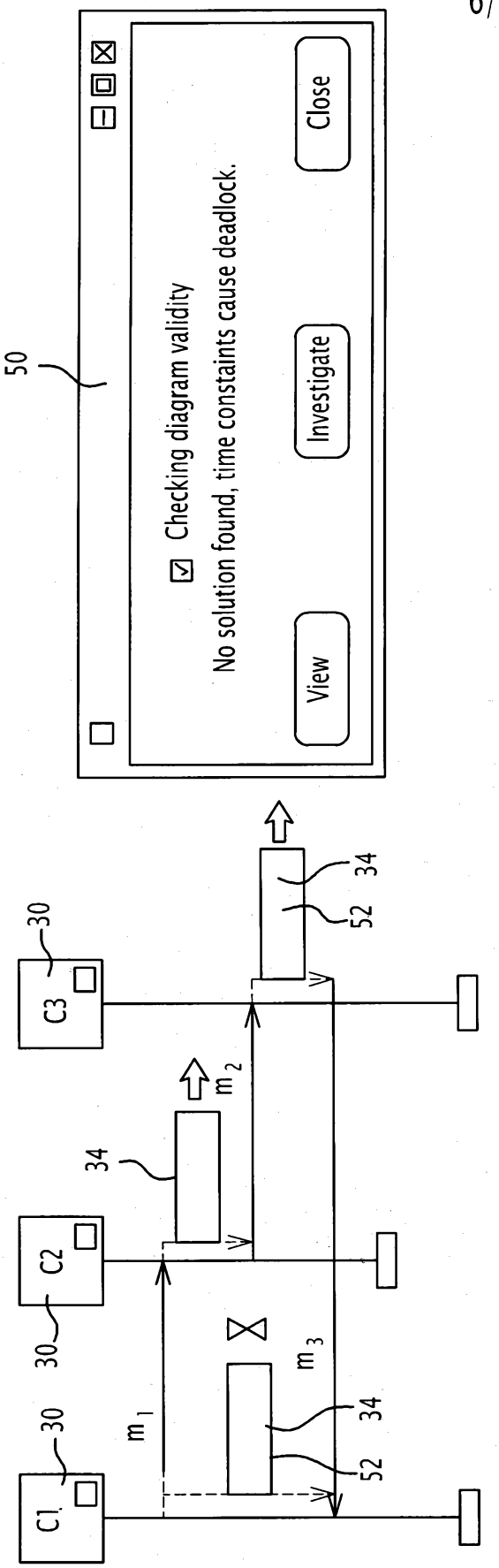
50
Checking diagram validity
No solution found, time constaints cause deadlock.
View
Investigate
Close
C1
C2
C3
30
34
52
m1
m2
m3

FIG.9

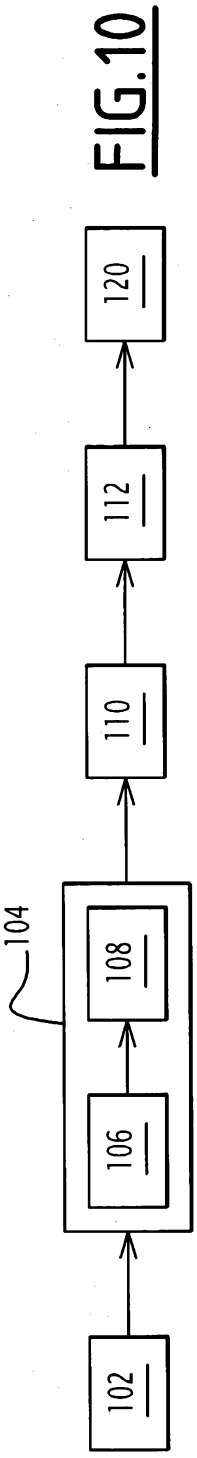
102
104
106
108
110
112
120

FIG.10

N° d'enregistrement national : 1500075 N° de publication : 3031820

RAPPORT DE RECHERCHE

articles L.612-14, L.612-17 et R.612-53 à 69 du code de la propriété intellectuelle

OBJET DU RAPPORT DE RECHERCHE

L'I.N.P.I. annexe à chaque brevet un "RAPPORT DE RECHERCHE" citant les éléments de l'état de la technique qui peuvent être pris en considération pour apprécier la brevetabilité de l'invention, au sens des articles L. 611-11 (nouveauté) et L. 611-14 (activité inventive) du code de la propriété intellectuelle. Ce rapport porte sur les revendications du brevet qui définissent l'objet de l'invention et délimitent l'étendue de la protection.

Après délivrance, l'I.N.P.I. peut, à la requête de toute personne intéressée, formuler un "AVIS DOCUMENTAIRE" sur la base des documents cités dans ce rapport de recherche et de tout autre document que le requérant souhaite voir prendre en considération.

CONDITIONS D'ÉTABLISSEMENT DU PRÉSENT RAPPORT DE RECHERCHE

- [x] Le demandeur a présenté des observations en réponse au rapport de recherche préliminaire.
- [x] Le demandeur a maintenu les revendications.
- [ ] Le demandeur a modifié les revendications.
- [ ] Le demandeur a modifié la description pour en éliminer les éléments qui n'étaient plus en concordance avec les nouvelles revendications.
- [ ] Les tiers ont présenté des observations après publication du rapport de recherche préliminaire.
- [ ] Un rapport de recherche préliminaire complémentaire a été établi.

DOCUMENTS CITÉS DANS LE PRÉSENT RAPPORT DE RECHERCHE

La répartition des documents entre les rubriques 1, 2 et 3 tient compte, le cas échéant, des revendications déposées en dernier lieu et/ou des observations présentées.

- [x] Les documents énumérés à la rubrique 1 ci-après sont susceptibles d'être pris en considération pour apprécier la brevetabilité de l'invention.
- [ ] Les documents énumérés à la rubrique 2 ci-après illustrent l'arrière-plan technologique général.
- [ ] Les documents énumérés à la rubrique 3 ci-après ont été cités en cours de procédure, mais leur pertinence dépend de la validité des priorités revendiquées.
- [ ] Aucun document n'a été cité en cours de procédure.

N° d'enregistrement national : 1500075 N° de publication : 3031820

1. ELEMENTS DE L'ETAT DE LA TECHNIQUE SUSCEPTIBLES D'ETRE PRIS EN CONSIDERATION POUR APPRECIER LA BREVETABILITE DE L'INVENTION

US 2013/263092 A1 (CHIKAHISA MASAKI [JP] ET AL)
3 octobre 2013 (2013-10-03)

US 6 516 306 B1 (ALUR RAJEEV [US] ET AL)
4 février 2003 (2003-02-04)

KANG I ET AL: "AN EFFICIENT STATE SPACE GENERATION FOR ANALYSIS OF REAL-TIME SYSTEMS", SOFTWARE ENGINEERING NOTES, ACM, NEW YORK, NY, US, vol. 21, no. 3, 1 mai 1996 (1996-05-01), pages 4-13, XP000584204,
ISSN: 0163-5948, DOI: 10.1145/226295.226297

2. ELEMENTS DE L'ETAT DE LA TECHNIQUE ILLUSTRANT L'ARRIERE-PLAN TECHNOLOGIQUE GENERAL

NEANT

3. ELEMENTS DE L'ETAT DE LA TECHNIQUE DONT LA PERTINENCE DEPEND DE LA VALIDITE DES PRIORITES

NEANT