(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

PARIS

(11) **N° de publication :** **2 910 656**

(à n'utiliser que pour les commandes de reproduction)

(21) **N° d'enregistrement national :** **06 11087**

(51) Int Cl[8] : ***G 06 F 11/28*** *(2006.01), G 06 F 9/455*

(12) DEMANDE DE BREVET D'INVENTION A1

(22) **Date de dépôt :** 20.12.06.

(30) **Priorité :**

(43) **Date de mise à la disposition du public de la demande :** 27.06.08 Bulletin 08/26.

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

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

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

(72) **Inventeur(s) :** SOUSSIEL OLIVIER et CAILLAUD CHRISTOPHE.

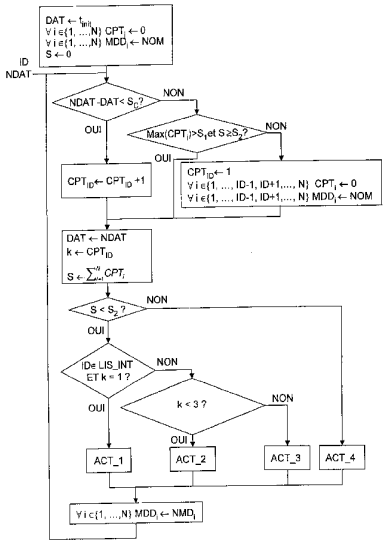
(73) **Titulaire(s) :**

(74) **Mandataire(s) :** MARKS & CLERK FRANCE.

(54) DISPOSITIF ET PROCEDE DE GESTION DE DEFAILLANCE DE TACHE DE PROCESSUS.

(57) Le domaine de l'invention est celui de la gestion de défaillance de tâche d'un processus. L'invention concerne un procédé de gestion de défaillance d'exécution de tâches AP_i d'un processus, le processus comportant un nombre de tâches égal à N, i désignant un indice identifiant les tâches et étant un nombre entier compris entre 1 et N, une exécution de la tâche AP_i étant démarrée suivant un mode de démarrage MDD_i.

Selon l'invention, le mode de démarrage des tâches AP_i du processus suite à une défaillance affectant une tâche AP_{ID} dépend d'un historique de défaillances qui a affecté chacune des tâches individuellement.



FR 2 910 656 - A1

Dispositif et procédé de gestion de défaillance de tâche de processus

Le domaine de l'invention est celui de la gestion de défaillance de tâche d'un processus.

L'invention concerne plus spécifiquement des processus complexes ayant une fonction critique comme par exemple un système de gestion de vol (connu aussi sous l'appellation anglaise « Flight Management System » ou l'acronyme FMS), embarqué à bord d'un aéronef.

En général, un processus ou une application logicielle complexe peut se décomposer en plusieurs tâches. Ces tâches s'exécutent indépendamment les unes des autres et disposent chacune d'un jeu de données locales propres à la tâche et d'un jeu de données communes partagées entre les tâches. Les tâches agissent sur ces différentes données, et possèdent en général plusieurs modes de fonctionnement qui correspondent à des algorithmes plus ou moins complexes, appelés respectivement mode nominal et modes dégradés.

Lorsqu'un processus assure une fonction critique, une défaillance d'une des tâches composant le processus peut entraîner une perte temporaire ou permanente de la totalité de la fonction du processus. Par exemple, pour un système de gestion de vol, FMS embarqué sur un aéronef, une exception logicielle ou une divergence de convergence affectant un algorithme de tracé de trajectoire est susceptible d'avoir des conséquences très graves sur la conduite de l'aéronef.

Le processus est en général conçu de façon à minimiser les conséquences des défaillances des tâches qui le composent. Cette minimisation peut être obtenue, d'une part en évitant que les défaillances surviennent, d'autre part, en prévoyant des mécanismes pour qu'après une détection d'une défaillance, la tâche défaillante et le processus soient replacés rapidement dans un état stable.

On évite que des défaillances affectent des tâches du processus en prenant des précautions particulièrement draconiennes lors de la conception des tâches du processus pour identifier des situations pouvant induire des défaillances.

On prévoit des mécanismes pour qu'une défaillance ne place pas le processus dans un état instable récurrent, pour ce faire, le mécanisme

consiste, par exemple, en une interruption de l'exécution de la tâche que l'on a détectée défaillante et un re-démarrage de l'exécution de cette tâche soit en mode dégradé soit en modifiant le jeu de données qu'elle utilise.

En raison de la quantité élevée d'information qu'un processus reçoit pendant son exécution, il est économiquement impossible d'envisager de manière exhaustive toutes les combinaisons de données présentées au processus lors des phases de conception, de codage et de test du processus. Par exemple, un FMS embarqué sur un aéronef concentre des données issues de senseurs pour la navigation (IRS acronyme de l'expression anglaise de « Inertial Reference System », GPS acronyme de l'expression anglaise « Global Positionning System », etc..), de données issues de bases de données de navigation pour élaborer le plan de vol électronique et sa trajectoire latérale de référence, de données issues de bases de données de performance pour élaborer les prédictions le long du plan de vol et enfin de données issues d'entrées manuelles venant de la part de l'équipage, en général pour initialiser les calculs, ou d'entrées automatiques par une liaison de donnée numérique sol/bord connue sous l'appellation anglaise «Datalink », venant de la compagnie aérienne qui exploite l'aéronef ou de centres de contrôles, on parle dans ce cas de « Air Traffic Control » ou on emploie l'acronyme anglais « ATC ». A cette combinaison de données il convient d'ajouter la combinaison des modes de fonctionnement des différentes tâches : soit au total une combinatoire si étendue qu'elle est impossible à envisager au cours de tests exhaustifs.

Pour sortir rapidement le processus d'un état instable dans lequel une défaillance d'une de ses tâches l'a placé, il est habituel de recourir à un dispositif de gestion des défaillance des tâches qui est intégré au système exécutant le processus.

La mission principale qui est dévolue à un tel dispositif de gestion de défaillance de tâche est d'éviter une perte totale, temporaire ou permanente, de la fonction du processus ou des données dont le processus est responsable. En effet, ce sont ces pertes totales qui entraînent les conséquences les plus graves : dans le cas du FMS, une perte temporaire ou une interruption de l'exécution de l'acquisition de la position GPS de l'aéronef par le FMS peut être tolérée, mais l'interruption simultanée de

toutes les tâches composant le FMS est très pénalisante pour un pilote d'aéronef.

Il est connu, dans l'art antérieur, des dispositifs de gestion de défaillance de tâche qui, lorsqu'une défaillance de la tâche est détectée interrompent sélectivement une ou plusieurs tâches du processus et démarrent une nouvelle exécution de ces tâches. Le démarrage de la nouvelle exécution de la tache est réalisé dans un mode de fonctionnement différent du mode de fonctionnement antérieur et/ou en employant un jeu de donné prédéfini différent de celui employé antérieurement. La détermination du mode de fonctionnement ou du jeu de donné employés suit une certaine logique.

La logique employée par les dispositifs de l'art antérieur est fondée le plus souvent sur un comptage d'un nombre de défaillances des tâches du processus. A la suite d'une détection de défaillance, une action correctrice est prise. Plus on a détecté un nombre important de défaillances du processus qui paraissent liées entre elles et plus sévère est l'effet de la mesure correctrice sur le fonctionnement du processus. Pour décrire les actions correctrices on définit habituellement différents types de démarrage d'exécution de tâche d'un processus qui succèdent à une interruption d'exécution :

- Un premier type de démarrage consiste à démarrer l'exécution de la tâche défaillante ou de l'ensemble des tâches du processus en employant un mode de fonctionnement nominal et un jeu de données identique à celui employé par la tâche lorsque la précédente exécution de la tâche a été interrompue ;
- Un deuxième type de démarrage consiste à démarrer l'exécution de l'ensemble des tâche du processus en employant un ou des jeux de données réinitialisés, le mode de fonctionnement des tâches du processus est le mode nominal ;
- Un troisième type de démarrage consiste à démarrer l'exécution de l'ensemble des tâche du processus en employant un mode de fonctionnement dit « dégradé » et un ou des jeux de données réinitialisés.

Un mode dégradé correspond à un mode de fonctionnement moins performant que le mode nominal, par exemple mettant en œuvre un

algorithme de complexité moins élevée que l'algorithme mis en œuvre dans le mode de fonctionnement nominal.

Le deuxième type de démarrage est en général considéré comme plaçant la tâche défaillante dans un état plus stable que celui auquel conduit un démarrage du premier type, mais il présente l'inconvénient d'occasionner une perte de données ;

Le troisième type de démarrage est en général considéré comme plaçant la tâche défaillante dans un état plus stable que celui auquel conduit un démarrage du deuxième type, mais il présente l'inconvénient d'occasionner une perte de données et de réduire les fonctions du processus.

Les dispositifs de l'art antérieur ont fortement réduit les occurrences de perte totale de la fonction des processus. Toutefois les dispositifs de gestion de défaillance de tâche d'un processus de l'art antérieur souffrent d'un certain nombre d'inconvénients.

Un premier inconvénient des procédés selon l'art antérieur réside dans le caractère global du comptage des défaillances affectant les tâches du processus qu'ils mettent en œuvre. Le caractère global du comptage ne permet pas de distinguer une situation dans laquelle toutes les tâches sont affectées plus ou moins aléatoirement d'une défaillance d'une situation dans laquelle une tâche particulière est affectée de défaillances répétées.

Un deuxième inconvénient, lié au premier inconvénient, vient de ce qu'en empêchant une identification d'une tâche particulière plus fragile que les autres, c'est à dire une identification d'une tâche plus fréquemment affectée d'une défaillance que les autres, les procédés de l'art antérieur interdisent également, de fait, de mener une analyse visant à déterminer l'origine des défaillances affectant cette tâche particulière. En effet, une fois qu'une tâche particulièrement défaillante est identifiée, il est possible d'investiguer pour déterminer si la défaillance est liée à son jeu de données ou à une instabilité de son mode de fonctionnement.

Cette investigation consiste par exemple à successivement interrompre l'exécution de la tâche défaillante puis à redémarrer cette exécution dans un mode de démarrage définissant un mode de

fonctionnement qui est dégradé par rapport à la précédente exécution, et/ou un jeu de donnée qui est réduit par rapport à la précédente exécution.

Par exemple, suite à une détection d'une défaillance affectant une tâche AP on a procédé à première interruption et un premier redémarrage de l'exécution de la tâche AP. Si on détecte une deuxième défaillance affectant cette tâche AP, et que la deuxième défaillance parait liée avec la première, on interrompt à nouveau puis on redémarre l'exécution de la tâche AP, mais cette fois avec un jeu de données différent.

Si par la suite, aucune défaillance n'affecte plus la tâche AP, on peut conclure que le jeu de donné était à l'origine de la défaillance, sinon, il est possible de continuer l'investigation en modifiant par la suite à nouveau le jeu de données ou encore le mode de fonctionnement.

Enfin, pour certains processus, les conséquences d'une perte d'un jeu de données, même momentanée, sont si graves que l'on cherche toujours à améliorer les performances des dispositifs de gestion de défaillance de tâche. En particulier, on cherche à éviter de perdre un jeu de données d'une tâche non défaillante en retardant l'application d'une action correctrice ultime qui consiste à réinitialiser les jeux de données de toutes les tâches du processus avant un ultime démarrage des tâches du processus. Dans le cas du FMS, on considère en effet que les données liées au plan de vol sont si sensibles qu'il est souhaitable de les conserver le plus longtemps possible.

L'objet de la présente invention est de pallier les inconvénients des dispositifs de gestion des défaillance de tâches de l'art antérieur pour augmenter la disponibilité d'un nombre maximum de tâches d'un processus lorsque des défaillances récurrentes touchent les tâches du processus.

Plus précisément l'invention a pour objet un procédé de gestion de défaillance d'exécution de tâches AP_i d'un processus, le processus comportant un nombre de tâches égal à N, i désignant un indice identifiant les tâches et étant un nombre entier compris entre 1 et N, une exécution de la tâche AP_i étant démarrée suivant un mode de démarrage MDD_i caractérisé en ce que le mode de démarrage des tâches AP_i du processus suite à une défaillance affectant une tâche AP_{ID} dépend d'un historique de défaillances qui a affecté chacune des tâches individuellement.

Un premier avantage du procédé selon l'invention tient en ce qu'il a la faculté de prend en compte une information de défaillance à l'échelle d'une tâche individuelle et non plus à l'échelle du processus. C'est à dire qu'une action correctrice appliquée par un procédé selon l'invention, suite à une détection de défaillance courante d'une tâche AP_{ID} a un effet sur les tâches AP_i qui peut être fonction de ce que :

- la défaillance courante affecte la tâche AP_{ID} ;
- la tâche AP_{ID} a, par le passé, été affectée par un nombre de défaillance égal à CPT_{ID} ;
- un précédent mode de démarrage de la tâche AP_i, dernier mode de démarrage en date, est le mode MDD_i.

Cette faculté permet de graduer l'effet des mesures correctrices : Considérons par exemple, une mesure correctrice prise suite à une détection d'une défaillance courante affectant la tâche AP_{ID} d'un processus. Cette mesure correctrice définit un mode de démarrage d'une tâche AP_i du processus qui est d'autant plus restrictif par rapport au mode de démarrage précédent de la tâche AP_i que :

- la tâche AP_{ID} est critique pour le processus,
- le nombre de défaillance ayant affectée la tâche AP_{ID} par le passé est élevé, et
- le nombre de démarrage effectué par la tâche AP_i est élevé.

Un deuxième avantage du procédé selon l'invention tient en ce qu'un jeu de données D_i d'une tâche AP_i qui est abandonné à la suite de l'application d'une action correctrice peut être réemployé lors d'une application d'une action correctrice ultérieure. En effet, les jeux de données des tâches AP_i sont stockés avant toute interruption d'une tâche en application d'une mesure correctrice. Il est avantageux de démarrer une exécution de tâche avec un jeu de données qui a été éprouvé lors d'une exécution antérieure.

L'invention concerne aussi un dispositif de gestion de défaillance de tâches AP_i d'un processus, ledit dispositif mettant en œuvre un procédé selon l'invention, ledit dispositif détectant une défaillance courante d'exécution affectant une tâche AP_{ID} du processus, la détection de la défaillance courante succédant à une détection antérieure

d'une défaillance, dite défaillance antérieure, ayant affecté une des tâches AP_i, caractérisé en ce qu'il comporte :

- une liste LIS_INT qui contient des indices de tâches AP_i dont une exécution peut être interrompue et démarrée individuellement sans perturber une exécution ou un démarrage d'une autre tâche du processus ;
- une table TAB qui contient des modes de démarrage de principe $MD_{i, ID, k}$ à employer pour démarrer la tâche AP_i, à la suite d'une défaillance courante affectant la tâche d'indice ID pour la k-ième fois.

L'invention concerne enfin, un système exécutant un processus comportant un nombre de tâches AP_i égal à N, i désignant un indice identifiant les tâches du processus et étant un nombre entier compris entre 1 et N, ledit système comportant au moins N unités de calcul UC_i exécutant chacune la tâche AP_i et un dispositif de gestion de défaillance de tâche AP_i d'un processus selon l'invention, caractérisé en ce que lorsqu'une tâche AP_{ID} est affectée par une défaillance courante, une date NDAT de détection de défaillance ainsi qu'un indice ID de tache défaillante sont délivrés au dispositif de gestion de défaillance et en ce que lorsque le système détecte qu'une défaillance courante d'exécution affecte une tâche AP_{ID}, elle produit une date NDAT de détection de défaillance et un indice ID de tâche défaillante à destination dudit dispositif.

D'autres caractéristiques et avantages de l'invention apparaîtront à la lecture de la description détaillée qui suit, faite à titre d'exemple non limitatif et en référence aux dessins annexés dans lesquels :

- la figure 1 représente schématiquement un système comportant trois unités de calcul UC_1, UC_2, UC_3, et un dispositif de gestion de défaillance de tâche ;
- la figure 2 représente schématiquement une architecture d'un dispositif de gestion de défaillance de tâche selon l'art antérieur ;
- la figure 3 représente un exemple d'organigramme d'un procédé de gestion de défaillance de tâche selon l'art antérieur ;
- la figure 4 représente schématiquement un dispositif de gestion de défaillance de tâche selon l'invention ;

- la figure 5 représente un exemple d'organigramme d'un procédé de gestion de défaillance de tâche selon l'invention.

D'une figure à l'autre, les mêmes éléments sont repérés par les mêmes références.

La figure 1 représente schématiquement un système PRO, 1, par exemple un FMS, exécutant un processus. Le système PRO, 1 comporte trois unités de calcul UC_1, 10, UC_2, 20, UC_3, 30 exécutant chacune, par exemple en parallèle, une tâche AP_1, AP_2, AP_3, et un dispositif de gestion de défaillance de tâche EH, 100 exécutant un procédé de gestion de défaillance de tâche selon l'art antérieur. Le dispositif de gestion de défaillance de tâche peut aussi être dénommé par l'appellation anglaise « Error Handler ».

Chaque tâche AP_1, AP_2, AP_3 est exécutée suivant un mode de fonctionnement qui lui est particulier et dispose d'un jeu de données qui lui est propre. Le jeu de données comporte des données locales qui sont stockées dans une mémoire volatile de l'unité de calcul UC_1, UC_2, UC_3 et des données communes qui sont utilisées par plusieurs tâches du système PRO 1, les données communes sont stockées dans une mémoire volatile du système PRO 1.

Dans un jeu de données, on distingue deux types de données :

- des données critiques, qui sont par exemple, pour un FMS embarqué sur un aéronef, des données de plan de vol communiquées par un pilote de l'aéronef ;

- des données non critiques, comme par exemple des paramètres de réglage des radionavigations.

Un mode de fonctionnement décrit par exemple un algorithme mis en oeuvre par une tâche pendant son exécution. La tâche possède au moins un mode de fonctionnement : un premier mode de fonctionnement, appelé mode de fonctionnement nominal, qui constitue l'algorithme optimal de la tâche et réalise toutes les fonctions assurées par la tâche. D'autres modes de fonctionnement de la tâche, appelés « modes dégradés », caractérisent des algorithmes qui comportent une ou des limitations par rapport au mode de fonctionnement nominal.

La figure 2 représente schématiquement un dispositif de gestion de défaillance de tâche EH, 100 selon l'art antérieur. Cette représentation permet d'expliquer comment fonctionne le dispositif de gestion de défaillance de tâche EH.

Le dispositif de gestion de défaillance de tâche EH, 100 est alerté lorsqu'une tâche AP_1, AP_2, AP_3 est défaillante. L'alerte de défaillance prend la forme d'une transmission d'un indice ID de tâche défaillante ainsi que d'une date de détection de défaillance NDAT.

Une tâche AP_1, AP_2, AP_3 peut détecter par ses propres moyens, qu'elle est défaillante, le système PRO, 1 peut également émettre une alerte de défaillance après avoir détecté une défaillance d'une des tâches. Dans les deux cas le dispositif de gestion de défaillance de tâche EH reçoit une alerte de défaillance comportant l'indice ID de tâche défaillante ainsi qu'une date de détection de défaillance courante NDAT.

Le dispositif de gestion de défaillance de tâche EH comporte un compteur de défaillance de tâches répertoriées CPT, 101 et un module de corrélation temporelle de défaillance TIM, 103.

Le compteur de défaillance de tâches répertoriées CPT comporte un nombre de défaillances d'exécution des tâches AP_i corrélées avec les défaillances précédentes ayant affecté des tâches du processus.

Le module de corrélation temporelle, TIM comporte notamment une date DAT de détection antérieure d'une défaillance d'une tâches AP_1, AP_2, AP_3.

Le compteur CPT et le dispositif de corrélation temporelle TIM sont initialisés au moment du démarrage du processus : une fois initialisés, le compteur CPT contient une valeur égale à 0 et la date DAT comporte une date de démarrage du processus t_{init}.

La **figure 3** représente un exemple d'organigramme d'un procédé de gestion de défaillance de tâche EH, 100 selon l'art antérieur.

Tout débute par une initialisation du compteur CPT et une initialisation du dispositif de corrélation temporelle TIM.

Par la suite, lorsqu'une détection courante d'une défaillance affectant une des tâches AP_1, AP_2, AP_3, a lieu à une date NDAT et que la détection courante succède à une détection antérieure qui a eu lieu à la date

DAT, on incrémente la valeur contenue dans le compteur CPT si et seulement si on détermine qu'une corrélation temporelle existe entre la défaillance courante et la défaillance antérieure, c'est à dire si et seulement si une durée séparant la date de détection courante NDAT et la date DAT de la détection antérieure est inférieure à un seuil de corrélation S_C prédéfini. Lorsqu'on détermine une absence de corrélation entre la défaillance courante et la défaillance antérieure, on substitue au contenu du compteur CPT une valeur égale à 1.

De cette façon, le procédé selon l'art antérieur distingue deux types de défaillances affectant des tâches du processus : une défaillance corrélée temporellement avec une défaillance antérieure ayant affecté des tâches du processus et une défaillance inopinée.

Une défaillance corrélée affecte une tâche du processus en lien avec une défaillance antérieure ayant également affecté une tâche du processus. Une défaillance courante est corrélée dans la mesure où la détection courante est espacée d'une date de détection d'une défaillance antérieure affectant une tâche du processus d'une durée inférieure à S_c.

Une défaillance inopinée affecte une tâche du processus de façon inopinée, c'est à dire sans rapport avec une défaillance antérieure affectant une tâche du processus.

Par exemple le seuil de corrélation Sc est égal à 1 minute. Lorsqu'une défaillance courante AP_i est détectée plus d'une minute après la détection antérieure, la défaillance courante est considérée comme non corrélée avec la défaillance antérieure.

Les actions correctrices AA_ACT_1, AA_ACT_2, AA_ACT_3, AA_ACT_4, AA_ACT_5, AA_ACT_6, ont un effet graduel sur le mode de fonctionnement des tâches.

Par exemple, lorsque une détection de défaillance affectant la tâche AP_{ID} est détectée, et que la valeur du compteur CPT vaut 1 ou 2, l'action correctrice AA_ACT_1 appliquée par le procédé selon l'art antérieur consiste à :

- interrompre l'exécution de la tâche AP_{ID}, puis à,
- démarrer l'exécution de la tâche AP_{ID}, suivant le mode de fonctionnement nominal, en conservant le jeu de donnée en cours au moment de l'interruption.

Lorsque une détection de défaillance affectant la tâche AP_{ID} est détectée, et que la valeur du compteur CPT vaut 3 ou 4, l'action correctrice AA_ACT_2 appliquée par le procédé selon l'art antérieur consiste à :

- interrompre l'exécution de toutes les tâches AP_i du processus, puis à,
- démarrer l'exécution de toutes les tâches AP_i suivant le mode de fonctionnement nominal, en conservant le jeu de donnée en cours au moment de l'interruption.

Lorsque une détection de défaillance affectant la tâche AP_{ID} est détectée, et que la valeur du compteur CPT vaut 5, l'action correctrice AA_ACT_3 appliquée par le procédé selon l'art antérieur consiste à :

- interrompre l'exécution de toutes les tâches AP_i du processus, puis à,
- démarrer l'exécution de toutes les tâches AP_i suivant le mode de fonctionnement nominal, en conservant une partie du jeu de donnée en cours au moment de l'interruption.

Lorsque une détection de défaillance affectant la tâche AP_{ID} est détectée, et que la valeur du compteur CPT vaut 6, l'action correctrice AA_ACT_4 appliquée par le procédé selon l'art antérieur consiste à :

- interrompre l'exécution de toutes les tâches AP_i du processus, puis à,
- démarrer l'exécution de toutes les tâches AP_i suivant le mode de fonctionnement nominal, en initialisant tous les jeux de donnée en cours au moment de l'interruption.

Enfin, lorsque une détection de défaillance affectant la tâche AP_{ID} est détectée, et que la valeur du compteur CPT est strictement supérieure à 6, l'action correctrice AA_ACT_5 appliquée par le procédé selon l'art antérieur consiste à interrompre l'exécution de toutes les tâches AP_i du processus.

La figure 4 représente schématiquement un dispositif de gestion de défaillance de tâche EH, 200 selon l'invention. Cette représentation permet d'expliquer comment fonctionne le dispositif de gestion de défaillance de tâche EH, 200 selon l'invention.

Le dispositif de gestion de défaillance de tâche EH, 200 détecte une défaillance courante d'exécution affectant une tâche AP_{ID} du processus. La détection de la défaillance courante succède à une détection antérieure d'une défaillance, dite défaillance antérieure, qui a affecté une des tâches AP_i du processus.

Avantageusement, le dispositif EH, comporte :

- une liste LIS_INT qui contient des indices de tâches AP_i dont une exécution peut être interrompue individuellement sans perturber une exécution d'une autre tâche du processus ;
- une table TAB qui contient des modes de démarrage de principe $MD_{i, ID, k}$ à employer pour démarrer la tâche AP_i, à la suite d'une défaillance courante affectant la tâche d'indice ID pour la k-ième fois.

Avantageusement, le dispositif EH, 200 comporte, en outre, une base de défaillances répertoriées, qui est mise à jour à chaque détection d'une défaillance courante affectant une tâche AP_i, ladite base de défaillances répertoriées comprend :

- des compteurs individuels CPT_i de défaillances de tâches AP_i, lesdits compteurs individuels CPT_i contenant un nombre de défaillances d'exécution des tâches AP_i corrélées avec les défaillances précédentes ;
- la date DAT de la détection antérieure ;
- un mode de démarrage MDD_i d'un précédent démarrage de la tâche APi, le précédent démarrage est le dernier démarrage en date de la tâche APi.

Avantageusement, le dispositif applique des actions correctrices ACT_1, ACT_2, ACT_3, ACT_4 ayant un effet graduel qui est fonction d'un contenu de la base de défaillances répertoriées mise à jour qui vise à interrompre puis à démarrer une exécution de tâches AP_i du processus suivant un mode de démarrage NMD_i.

L'invention concerne également un système PRO, 1 exécutant un processus comportant un nombre de tâches AP_i égal à N.

Le système PRO comporte au moins N unités de calcul UC_i exécutant chacune une tâche AP_i et un dispositif de gestion de défaillance de tâche EH, 200 selon l'invention. i désigne un indice identifiant les tâches du processus et est un nombre entier compris entre 1 et N.

Selon l'invention, les unités de calcul UC_i peuvent commander une sauvegarde totale ou partielle d'un jeu de données d'une unité de calcul UC_i distincte d'elles mêmes, dans certaines situations, à une fin de réutilisation ultérieure.

Par exemple lorsqu'une unité de calcul UC_1 reçoit une partie d'un jeu de données d'une unité de calcul UC_2 et que l'unité UC_1 a pu vérifier l'intégrité de ces données, l'unité de calcul UC_1 peut commander une sauvegarde de la partie du jeu de données que lui a transmis l'unité de calcul UC_2. La partie du jeu de données qui sauvegardée concerne en général des données critiques de l'unité de calcul UC_2, mais il est possible que la sauvegarde contienne également des données non-critiques.

Cette sauvegarde est particulièrement utile car elle permet de conserver des jeux de données, en totalité ou en partie, dont une unité de calcul a éprouvé la validité. Ces jeux de données sont présumés stables et peuvent être utilisés au cours de démarrages ultérieurs de la tâche.

Avantageusement, lorsqu'une première unité de calcul UC_i d'un système PRO selon l'invention transmet une partie du contenu du jeu de données D_i de la tâche AP_i qu'il exécute, à une deuxième unité de calcul UC_j du système PRO selon l'invention, où j est un indice différent de i, la deuxième unité UC_j est capable de commander une sauvegarde de la partie du contenu du jeu de données D_i qui lui a été transmis.

La figure 5 représente un exemple d'organigramme d'un procédé de gestion de défaillance de tâche selon l'invention.

Considérons un processus comportant un nombre de tâches égal à N, i désignant un indice identifiant les tâches et étant un nombre entier compris entre 1 et N.

Avantageusement, le mode de démarrage MDD_i définissant de façon unique un mode de fonctionnement de la tâche AP_i ainsi qu'un contenu d'un jeu de données D_i à employer au démarrage de l'exécution de la tâche AP_i, une détection d'une défaillance courante d'exécution affectant une tâche AP_{ID} produisant une date NDAT de détection de défaillance et un indice ID de tâche défaillante, la détection de la défaillance courante succédant à une détection antérieure d'une défaillance, dite défaillance antérieure, ayant

affecté une des tâches AP_i, ladite détection antérieure étant réalisée à une date DAT, caractérisé en ce qu'il comporte les étapes suivantes :

- Initialiser une base de défaillances répertoriées qui comporte :
  - des compteurs individuels CPT_i de défaillances de tâches AP_i, lesdits compteurs individuels CPT_i contenant un nombre de défaillances d'exécution des tâches AP_i corrélées avec les défaillances précédentes ;
  - la date DAT de la détection antérieure ;
  - un mode de démarrage MDD_i d'un précédent démarrage de la tâche AP_i, le précédent démarrage est le dernier démarrage en date de la tâche AP_i;- un cumul S égale à une somme des valeurs des compteurs individuels de défaillance de tâches CPT_i, pour tous les indices i.
- Lire un contenu de la liste LIS_INT ;
- Lorsque la défaillance d'exécution de la tâche AP_{ID} est détectée, mettre à jour la base de défaillances répertoriées ;
- Appliquer une action correctrice (ACT_1, ACT_2, ACT_3, ACT_4) qui a un effet sur l'exécution des tâches AP_i, l'action correctrice appliquée (ACT_1, ACT_2, ACT_3, ACT_4) est fonction d'un contenu de la base de défaillances répertoriées mise à jour ;
- Lorsque l'effet de l'action correctrice appliquée (ACT_1, ACT_2, ACT_3, ACT_4) a conduit à interrompre puis à démarrer une tâche AP_i suivant un mode de démarrage assigné NMD_i, substituer le mode de démarrage assigné NMD_i au mode de démarrage MDD_i, pour tous les indices i.

La liste LIS_INT contient des indices de tâches AP_i dont une exécution peut être interrompue individuellement sans perturber une exécution d'une autre tâche du processus.

L'exécution de la tâche AP_i est démarrée suivant un mode de démarrage MDD_i, le mode de démarrage MDD_i définissant de façon unique un mode de fonctionnement de la tâche AP_i ainsi qu'un contenu d'un jeu de données D_i à employer au démarrage de l'exécution de la tâche AP_i.

Une détection d'une défaillance courante d'exécution affectant une tâche AP_{ID} caractérisé par une date NDAT de détection de défaillance et un indice ID de tâche défaillante.

La détection de la défaillance courante succède à une détection antérieure d'une défaillance, dite défaillance antérieure, qui a affecté une des tâches AP_i, ladite détection antérieure est réalisée à une date DAT,

Une première étape du procédé selon l'invention consiste à initialiser la base de défaillances répertoriées.

Avantageusement, l'initialisation de la base de défaillances répertoriées comporte les étapes suivantes :

- Initialiser les compteurs individuels CPT_i, une fois initialisés les compteurs individuels CPT_i contiennent une valeur égale à 0, pour tous les indices i ;
- Initialiser la date DAT de la détection antérieure, une fois initialisée, la date DAT comporte une date de démarrage du processus t_{init} ;
- Initialiser les mode de démarrage MDD_i, pour tous les indices i, une fois initialisées, les mode de démarrage MDD_i comportent un mode de démarrage nominal NOM qui correspond à un mode de fonctionnement optimal de la tâche AP_i ;
- Initialiser le cumul S : une fois initialisé le cumul S contient une valeur égale à 0 ;

Une deuxième étape du procédé selon l'invention consiste en une lecture d'un contenu de la liste LIS_INT, pour que le dispositif prenne connaissance des tâches dont l'exécution est susceptible d'être interrompue et démarrée individuellement, sans perturber une exécution d'une autre tâche du processus

Une troisième étape du procédé selon l'invention consiste en une mise à jour de la base de défaillances répertoriées.

Avantageusement, cette mise à jour d'une base de défaillances répertoriées comporte les étapes suivantes :

- Déterminer une valeur maximale M des compteurs individuels CPT_i pour tous les indices i;
- Déterminer une existence de corrélation entre la défaillance courante et la défaillance antérieure ;

- Lorsqu'on détermine l'existence d'une corrélation entre la défaillance courante et la défaillance antérieure, incrémenter la valeur contenue dans le compteur individuel CPT_{ID} ;

- Lorsqu'on détermine une absence de corrélation entre la défaillance courante et la défaillance antérieure, lorsque la valeur maximale M est inférieure ou égale à un premier seuil S_1, et lorsque le cumul S est strictement supérieur à un deuxième seuil S_2, substituer à un contenu du compteur individuel CPT_{ID} une valeur égale à 1, et initialiser les compteurs individuels CPT_i, pour tous les indices i différents de ID ;

- Substituer la date de détection courante NDAT à la date DAT de la détection antérieure ;

- Déterminer un mode de démarrage de principe $MD_{i, ID, k}$ pour la tâche AP_i, pour tous les indices i, en fonction de l'indice ID de la tâche affectée par la défaillance courante et d'une valeur k, où k est égal à une valeur contenue dans le compteur individuel CPT_{ID} ;

- Déterminer un cumul S égale à une somme des valeurs des compteurs individuels de défaillance de tâches CPT_i, pour tous les indices i ;

- Déterminer l'action correctrice à appliquer (ACT_1, ACT_2, ACT_3, ACT_4) en fonction d'une comparaison du cumul S avec le deuxième seuil S_2, de k et d'une appartenance de l'indice ID à la liste LIS_INT ;

- Déterminer le mode de démarrage assigné NMD_i à la tâche AP_i par l'action correctrice à appliquer (ACT_1, ACT_2, ACT_3, ACT_4), pour tous les indices i.

Avantageusement, la détermination d'une existence de corrélation entre la défaillance courante et la défaillance antérieure est fondée sur une comparaison entre une durée séparant la date de détection courante NDAT et la date DAT de la détection antérieure et un seuil de corrélation S_C.

Avantageusement, la détermination d'un mode de démarrage de principe $MD_{i,ID, k}$ pour la tâche AP_i, à la suite d'une défaillance affectant la tâche d'indice ID pour la k-ième fois, consiste à lire une information contenue dans la table prédéfinie TAB.

Une quatrième étape du procédé selon l'invention consiste en une application d'une action correctrice (ACT_1, ACT_2, ACT_3, ACT_4) qui a

un effet sur l'exécution des tâches AP_i. L'effet de l'action correctrice appliquée est fonction d'un contenu de la base de défaillances répertoriées mise à jour.

Avantageusement, une action correctrice appliquée (ACT_1, ACT_2, ACT_3, ACT_4) comporte une première étape de sauvegarde des jeux de données D_i des tâches AP_i, pour tous les indices i.

Avantageusement, lorsque le cumul S est supérieur ou égal au deuxième seuil S_2, une action correctrice ACT_4 est appliquée qui comporte, en outre, les étapes suivantes, pour tous les indices i :

- Interrompre l'exécution de la tâche AP_i ;
- Démarrer l'exécution de la tâche AP_i, suivant un mode de démarrage NMD_i déterminé en fonction de la valeur du cumul S.

Avantageusement, lorsque la valeur du cumul S est supérieure ou égale à S_2+2, le mode de démarrage NMD_i correspond à une interruption permanente d'exécution des tâches AP_i.

La liste LIS_INT contient des indices de tâches AP_i dont une exécution peut être interrompue et démarrée individuellement sans perturber l'exécution d'une autre tâche du processus.

Avantageusement, lorsque le cumul S est strictement inférieur à S_2, k est égal à 1 et l'indice ID fait partie de la liste LIS_INT, une action correctrice ACT_1 est appliquée qui comporte, en outre, les étapes suivantes:

- Interrompre l'exécution de la tache AP_{ID} ;
- Démarrer l'exécution de la tache AP_{ID} suivant un mode de démarrage NMD_{ID} identique au mode de démarrage MDD_{ID} du précédent démarrage de la tache AP_{ID}.

Avantageusement, lorsque le cumul S est strictement inférieur au deuxième seuil S_2 et lorsque k est différent de 1 ou l'indice ID ne fait pas partie de la liste LIS_INT, et lorsque k est strictement inférieur à 3, une action correctrice ACT_2 est appliquée qui comporte, en outre, les étapes suivantes, pour tous les indices i :

- Interrompre l'exécution de la tâche AP_i ;
- Démarrer l'exécution de la tâche AP_i suivant un mode de démarrage NMD_i qui est identique au mode de démarrage MDD_i du précédent démarrage de la tâche AP_i.

Avantageusement, lorsque le cumul S est strictement inférieur au deuxième seuil S_2 et lorsque k est différent de 1 ou l'indice ID ne fait pas partie de la liste LIS_INT et lorsque k est supérieur ou égal à 3, une action correctrice ACT_3 est appliquée qui comporte, en outre, les étapes suivantes, pour tous les indices i :

- Interrompre l'exécution de la tâche AP_i ;
- Démarrer l'exécution de la tâche AP_i, suivant un mode de démarrage NMD_i déterminé à partir d'une comparaison entre le mode de démarrage MDD_i du précédent démarrage de la tâche AP_i et le mode de démarrage de principe $MD_{i, ID, k}$.

Avantageusement, un mode de démarrage NMD_i d'une tâche AP_i est un nombre entier et en ce que plus une valeur du mode de démarrage NMD_i est élevée et plus une différence de fonction entre une exécution de la tâche AP_i démarrée suivant le mode de démarrage NMD_i et une exécution de la tâche AP_i démarrée suivant le mode de démarrage nominal est grande.

Avantageusement, le mode de démarrage nominal NOM vaut 0, et en ce que la détermination du mode de démarrage NMD_i consiste à affecter au mode de démarrage NMD_i une valeur égale au maximum entre la valeur du mode de démarrage MDD_i et la valeur du mode de démarrage de principe $MD_{i, ID, k}$.

Avantageusement, un mode de démarrage de principe $MD_{i, ID, k}$ définit un contenu de jeu de données D_i à employer au démarrage de l'exécution de la tâche AP_i qui correspond à un jeu de données sauvegardé.

Une cinquième étape du procédé selon l'invention consiste en une substitution du mode de démarrage assigné NMD_i au mode de démarrage MDD_i, pour tous les indices i, lorsque l'effet de l'action correctrice appliquée a conduit à interrompre puis à démarrer une tâche AP_i suivant un mode de démarrage assigné NMD_i.

Par ailleurs, un système PRO, 1 qui exécute un processus comportant un nombre de tâches AP_i égal à N et qui comporte au moins N unités de calcul UC_i exécutant chacune la tâche AP_i et un dispositif de gestion de défaillance de tâche AP_i du processus selon l'invention, a un

fonctionnement qui peut interférer avec l'organigramme présenté sur la figure 5.

Des événements extérieurs à un système PRO, 1 exécutant un processus, sont susceptibles de produire une modification substantielle du jeu de données de certaines tâches composant le processus.

Pour certains évènements bien identifiés, cette modification substantielle de jeu de données est telle qu'elle modifie fondamentalement l'état des tâches et même affecte l'état du processus dans son ensemble. Il est des situations où les modifications substantielles ont un effet positif sur la stabilité des taches concernées, c'est à dire que ces modifications placent la tâche concernée dans un état plus stable que celui dans lequel elle était.

Pour prendre en compte les effets de ces modifications substantielles de jeux de données particulières, le système PRO associe à une détection de certains évènements extérieurs au système une mise à jour de la base de défaillances répertoriées de son dispositif de gestion de défaillance de tâche.

Avantageusement, le système PRO selon l'invention comporte des moyens pour détecter des événements extérieurs au système EV, et une mise à jour de la base de défaillances répertoriées du dispositif de gestion de défaillance de tâche est déclenchée par une détection d'un événement extérieur au système EV.

Pour processus tel qu'un système de gestion de vol FMS équipant un aéronef, un mouvement de l'aéronef est un exemple d'événement extérieur.

Considérons en effet, une tâche AP_0 du FMS réalisant un tracé du plan de vol à partir de WAY_POINT entrés par un pilote de l'aéronef. Un jeu de données de la tache AP_0 comportant des WAY_POINT utiles pour tracer le plan de vol est modifié par le déplacement de l'aéronef lorsque l'aéronef a dépassé un des WAY_POINT. Si la tâche AP_0 était affectée par une série de défaillances successives, il est possible que la modification du jeu de données induite par le déplacement de l'aéronef est suffisante pour placer la tâche AP_0 hors d'un contexte produisant la série de défaillances. La mise à jour de la base de données de défaillances répertoriées du dispositif de gestion de défaillance est effectuée pour refléter ce changement d'état.

La mise à jour est prédéfinie par un concepteur du système PRO. Selon l'événement extérieur EV détecté, la mise à jour affecte des valeurs contenues dans des compteurs individuels CPT_i de certaines tâches prédéfinies.

Avantageusement, la mise à jour de la base de défaillances répertoriées comporte une étape d'initialisation des compteurs individuels CPT_i pour des tâches dont les indices sont rangés dans une liste L_1 qui est fonction de l'événement extérieur au système EV détecté par le système.

Selon l'événement extérieur EV détecté, la mise à jour affecte les valeurs des mode de démarrage MDD_i d'un précédent démarrage de certaines tâches prédéfinies.

Avantageusement, la mise à jour de la base de défaillances répertoriées comporte une étape d'initialisation des mode de démarrage MDD_i d'un précédent démarrage pour des tâches dont les indices sont rangés dans une liste L_2 qui est fonction de l'événement extérieur au système EV détecté par le système.

REVENDICATIONS

1. Procédé de gestion de défaillance d'exécution de tâches AP_i d'un processus, le processus comportant un nombre de tâches égal à N, i désignant un indice identifiant les tâches et étant un nombre entier compris entre 1 et N, une exécution de la tâche AP_i étant démarrée suivant un mode de démarrage MDD_i,
caractérisé en ce que le mode de démarrage MDD_i des tâches AP_i du processus suite à une défaillance affectant une tâche AP_{ID} dépend d'un historique des défaillances qui ont affecté chacune des tâches individuellement.

2. Procédé selon la revendication 1, le mode de démarrage MDD_i définissant de façon unique un mode de fonctionnement de la tâche AP_i ainsi qu'un contenu d'un jeu de données D_i à employer au démarrage de l'exécution de la tâche AP_i, une détection d'une défaillance courante d'exécution affectant une tâche AP_{ID} produisant une date NDAT de détection de défaillance et un indice ID de tâche défaillante, la détection de la défaillance courante succédant à une détection antérieure d'une défaillance, dite défaillance antérieure, ayant affecté une des tâches AP_i, ladite détection antérieure étant réalisée à une date DAT,
caractérisé en ce qu'il comporte les étapes suivantes :

- Initialiser une base de défaillances répertoriées qui comporte :
  - des compteurs individuels CPT_i de défaillances de tâches AP_i, lesdits compteurs individuels CPT_i contenant un nombre de défaillances d'exécution des tâches AP_i corrélées avec les défaillances précédentes ;
  - la date DAT de la détection antérieure ;- un mode de démarrage MDD_i d'un précédent démarrage de la tâche AP_i, le précédent démarrage est le dernier démarrage en date de la tâche AP_i ;- un cumul S égale à une somme des valeurs des compteurs individuels de défaillance de tâches CPT_i, pour tous les indices i.
- Lire un contenu d'une liste LIS_INT ;
- Lorsque la défaillance d'exécution de la tâche AP_{ID} est détectée, mettre à jour la base de défaillances répertoriées ;

- Appliquer une action correctrice (ACT_1, ACT_2, ACT_3, ACT_4) qui a un effet sur l'exécution des tâches AP_i, l'action correctrice appliquée (ACT_1, ACT_2, ACT_3, ACT_4) est fonction d'un contenu de la base de défaillances répertoriées mise à jour ;

- Lorsque l'effet de l'action correctrice appliquée (ACT_1, ACT_2, ACT_3, ACT_4) a conduit à interrompre puis à démarrer une tâche AP_i suivant un mode de démarrage assigné NMD_i, substituer le mode de démarrage assigné NMD_i au mode de démarrage MDD_i, pour tous les indices i.

3. Procédé selon la revendication 2 caractérisé en ce que l'initialisation de la base de défaillances répertoriées comporte les étapes suivantes :

- Initialiser les compteurs individuels CPT_i, une fois initialisés les compteurs individuels CPT_i contiennent une valeur égale à 0, pour tous les indices i ;

- Initialiser la date DAT de la détection antérieure, une fois initialisée, la date DAT comporte une date de démarrage du processus t_{init} ;

- Initialiser les mode de démarrage MDD_i, pour tous les indices i, une fois initialisées, les mode de démarrage MDD_i comportent un mode de démarrage nominal NOM qui correspond à un mode de fonctionnement optimal de la tâche AP_i ;

- Initialiser le cumul S : une fois initialisé le cumul S contient une valeur égale à 0 ;

4. Procédé selon l'une des revendications 2 ou 3, caractérisé en ce que la mise à jour d'une base de défaillances répertoriées comporte les étapes suivantes :

- Déterminer une valeur maximale M des compteurs individuels CPT_i pour tous les indices i;

- Déterminer une existence de corrélation entre la défaillance courante et la défaillance antérieure ;

- Lorsqu'on détermine l'existence d'une corrélation entre la défaillance courante et la défaillance antérieure, incrémenter la valeur contenue dans le compteur individuel CPT_{ID} ;

- Lorsqu'on détermine une absence de corrélation entre la défaillance courante et la défaillance antérieure, lorsque la valeur maximale M est inférieure ou égale à un premier seuil S_1, et lorsque le cumul S est strictement supérieur à un deuxième seuil S_2, substituer à un contenu du compteur individuel CPT_{ID} une valeur égale à 1, et initialiser les compteurs individuels CPT_i, pour tous les indices i différents de ID ;

- Substituer la date de détection courante NDAT à la date DAT de la détection antérieure ;

- Déterminer un mode de démarrage de principe $MD_{i, ID, k}$ pour la tâche AP_i, pour tous les indices i, en fonction de l'indice ID de la tâche affectée par la défaillance courante et d'une valeur k, où k est égal à une valeur contenue dans le compteur individuel CPT_{ID} ;

- Déterminer un cumul S égale à une somme des valeurs des compteurs individuels de défaillance de tâches CPT_i, pour tous les indices i ;

- Déterminer l'action correctrice à appliquer (ACT_1, ACT_2, ACT_3, ACT_4) en fonction d'une comparaison du cumul S avec le deuxième seuil S_2, de k et d'une appartenance de l'indice ID à la liste LIS_INT ;

- Déterminer le mode de démarrage assigné NMD_i à la tâche AP_i par l'action correctrice à appliquer (ACT_1, ACT_2, ACT_3, ACT_4), pour tous les indices i.

5. Procédé selon la revendication 4, caractérisé en ce que la détermination d'une existence de corrélation entre la défaillance courante et la défaillance antérieure est fondée sur une comparaison entre une durée séparant la date de détection courante NDAT et la date DAT de la détection antérieure et un seuil de corrélation S_C.

6. Procédé selon la revendication 4, caractérisé en ce que la détermination d'un mode de démarrage de principe $MD_{i,ID, k}$ pour la tâche AP_i, à la suite d'une défaillance affectant la tâche d'indice ID pour la k-ième fois, consiste à lire une information contenue dans une table prédéfinie TAB.

7. Procédé selon l'une des revendications 4 à 6 caractérisé en ce qu'une action correctrice appliquée (ACT_1, ACT_2, ACT_3, ACT_4) comporte une

première étape de sauvegarde des jeux de données D_i des tâches AP_i, pour tous les indices i.

8. Procédé selon la revendication 7, la liste LIS_INT contenant des indices de tâches AP_i dont une exécution peut être interrompue et démarrée individuellement sans perturber l'exécution d'une autre tâche du processus, caractérisé en ce que, lorsque le cumul S est strictement inférieur au deuxième seuil S_2, k est égal à 1 et l'indice ID fait partie de la liste LIS_INT, une action correctrice ACT_1 est appliquée qui comporte, en outre, les étapes suivantes :

- Interrompre l'exécution de la tache AP_{ID} ;
- Démarrer l'exécution de la tache AP_{ID} suivant un mode de démarrage NMD_{ID} identique au mode de démarrage MDD_{ID} du précédent démarrage de la tache AP_{ID}.

9. Procédé selon la revendication 7, caractérisé en ce que, lorsque le cumul S est strictement inférieur au deuxième seuil S_2 et lorsque k est différent de 1 ou l'indice ID ne fait pas partie de la liste LIS_INT, et lorsque k est strictement inférieur à 3, une action correctrice ACT_2 est appliquée qui comporte, en outre, les étapes suivantes, pour tous les indices i :

- Interrompre l'exécution de la tâche AP_i ;
- Démarrer l'exécution de la tâche AP_i suivant un mode de démarrage NMD_i qui est identique au mode de démarrage MDD_i du précédent démarrage de la tâche AP_i.

10. Procédé selon la revendication 7, caractérisé en ce que, lorsque le cumul S est strictement inférieur au deuxième seuil S_2 et lorsque k est différent de 1 ou l'indice ID ne fait pas partie de la liste LIS_INT et lorsque k est supérieur ou égal à 3, une action correctrice ACT_3 est appliquée qui comporte, en outre, les étapes suivantes, pour tous les indices i :

- Interrompre l'exécution de la tâche AP_i ;
- Démarrer l'exécution de la tâche AP_i, suivant un mode de démarrage NMD_i déterminé à partir d'une comparaison entre le mode de démarrage MDD_i du précédent démarrage de la tâche AP_i et le mode de démarrage de principe $MD_{i, ID, k}$.

11. Procédé selon la revendication 10, caractérisé en ce qu'un mode de démarrage NMD_i d'une tâche AP_i est un nombre entier et en ce que plus une valeur du mode de démarrage NMD_i est élevée et plus une différence de fonction entre une exécution de la tâche AP_i démarrée suivant le mode de démarrage NMD_i et une exécution de la tâche AP_i démarrée suivant le mode de démarrage nominal est grande.

12. Procédé selon la revendication 11, caractérisé en ce le mode de démarrage nominal NOM vaut 0, et en ce que la détermination du mode de démarrage NMD_i consiste à affecter au mode de démarrage NMD_i une valeur égale au maximum entre la valeur du mode de démarrage MDD_i et la valeur du mode de démarrage de principe $MD_{i, ID, k}$.

13. Procédé selon la revendication 7 caractérisé en ce que, lorsque le cumul S est supérieur ou égal au deuxième seuil S_2, une action correctrice ACT_4 est appliquée qui comporte, en outre, les étapes suivantes, pour tous les indices i :

- Interrompre l'exécution de la tâche AP_i ;
- Démarrer l'exécution de la tâche AP_i, suivant un mode de démarrage NMD_i déterminé en fonction de la valeur du cumul S.

14. Procédé selon la revendication 13, caractérisé en ce que, lorsque la valeur du cumul S est supérieure ou égale à S_2+2, le mode de démarrage NMD_i correspond à une interruption permanente d'exécution des tâches AP_i.

15. Procédé selon l'une des revendications 7 à 14, caractérisé en ce qu'un mode de démarrage de principe $MD_{i, ID, k}$ définit un contenu de jeu de données D_i à employer au démarrage de l'exécution de la tâche AP_i qui correspond à un jeu de données sauvegardé.

16. Dispositif de gestion de défaillance de tâches AP_i d'un processus, ledit dispositif mettant en œuvre un procédé selon l'une des revendications précédentes, ledit dispositif détectant une défaillance courante d'exécution affectant une tâche AP_{ID} du processus, la détection de la défaillance

courante succédant à une détection antérieure d'une défaillance, dite défaillance antérieure, ayant affecté une des tâches AP_i,

caractérisé en ce qu'il comporte :

- une liste LIS_INT qui contient des indices de tâches AP_i dont une exécution peut être interrompue individuellement sans perturber une exécution d'une autre tâche du processus ;

- une table TAB qui contient des modes de démarrage de principe $MD_{i, ID, k}$ à employer pour démarrer la tâche AP_i, à la suite d'une défaillance courante affectant la tâche d'indice ID pour la k-ième fois.

17. Dispositif selon la revendication précédente caractérisé en ce qu'il comporte, en outre, une base de défaillances répertoriées, qui est mise à jour à chaque détection d'une défaillance courante affectant une tâche AP_i, ladite base de défaillances répertoriées comprend :

- des compteurs individuels CPT_i de défaillances de tâches AP_i, lesdits compteurs individuels CPT_i contenant un nombre de défaillances d'exécution des tâches AP_i corrélées avec les défaillances précédentes ;

- la date DAT de la détection antérieure ;

- un mode de démarrage MDD_i d'un précédent démarrage de la tâche APi, le précédent démarrage est le dernier démarrage en date de la tâche APi.

et en ce que ledit dispositif applique des actions correctrices (ACT_1, ACT_2, ACT_3, ACT_4) ayant un effet graduel qui est fonction d'un contenu de la base de défaillances répertoriées mise à jour, l'effet graduel vise à interrompre puis à démarrer une exécution de tâches AP_i du processus, suivant un mode de démarrage NMD_i.

18. Système exécutant un processus comportant un nombre de tâches AP_i égal à N, i désignant un indice identifiant les tâches du processus et étant un nombre entier compris entre 1 et N, ledit système comportant au moins N unités de calcul UC_i exécutant chacune la tâche AP_i et un dispositif de gestion de défaillance de tâche AP_i d'un processus selon l'une des revendications 16 à 17,

caractérisé en ce en ce que, lorsqu'une tâche AP_{ID} est affectée par une défaillance courante, une date NDAT de détection de défaillance ainsi qu'un

indice ID de tache défaillante sont délivrés au dispositif de gestion de défaillance et en ce que lorsque le système détecte qu'une défaillance courante d'exécution affecte une tâche AP_{ID}, elle produit une date NDAT de détection de défaillance et un indice ID de tâche défaillante à destination dudit dispositif.

19. Système selon la revendication 18, caractérisé en ce que lorsqu'une première unité de calcul UC_i du système transmet une partie du contenu du jeu de données D_i de la tâche AP_i qu'il exécute, à une deuxième unité de calcul UC_j du système, où j est un indice différent de i, la deuxième unité UC_j est capable de commander une sauvegarde de la partie du contenu du jeu de données D_i qui lui a été transmis.

20. Système selon l'une des revendications 18 à 19, caractérisé en ce qu'il comporte des moyens pour détecter des événements extérieurs au système EV, et en ce qu'une mise à jour de la base de défaillances répertoriées du dispositif de gestion de défaillance de tâche est déclenchée par une détection d'un événement extérieur au système EV.

21. Système selon la revendication 20, caractérisé en ce que la mise à jour de la base de défaillances répertoriées comporte une étape d'initialisation des compteurs individuels CPT_i pour des tâches dont les indices sont rangés dans une liste L_1 qui est fonction de l'événement extérieur au système EV détecté par le système.

22. Système selon la revendication 20, caractérisé en ce que la mise à jour de la base de défaillances répertoriées comporte une étape d'initialisation des mode de démarrage MDD_i d'un précédent démarrage pour des tâches dont les indices sont rangés dans une liste L_2 qui est fonction de l'événement extérieur au système EV détecté par le système.

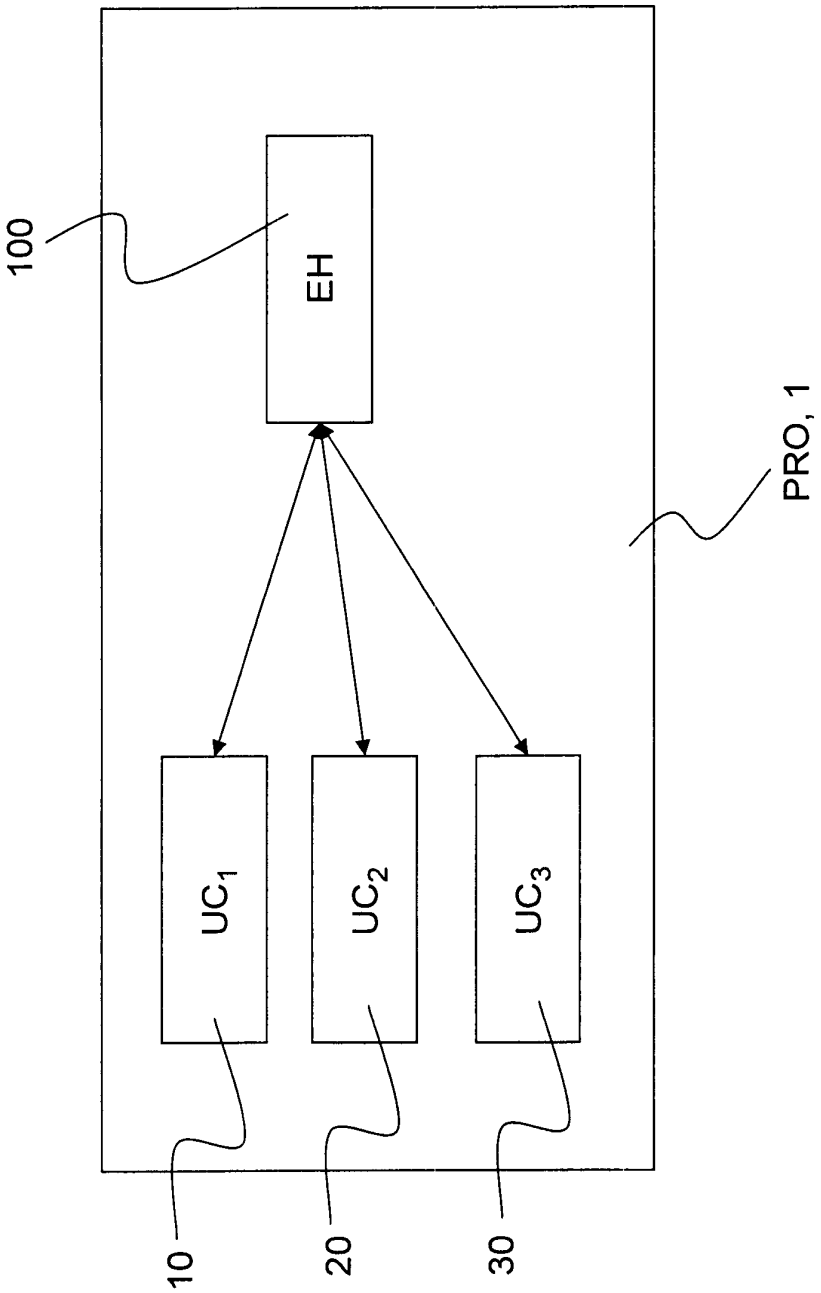


FIG.1

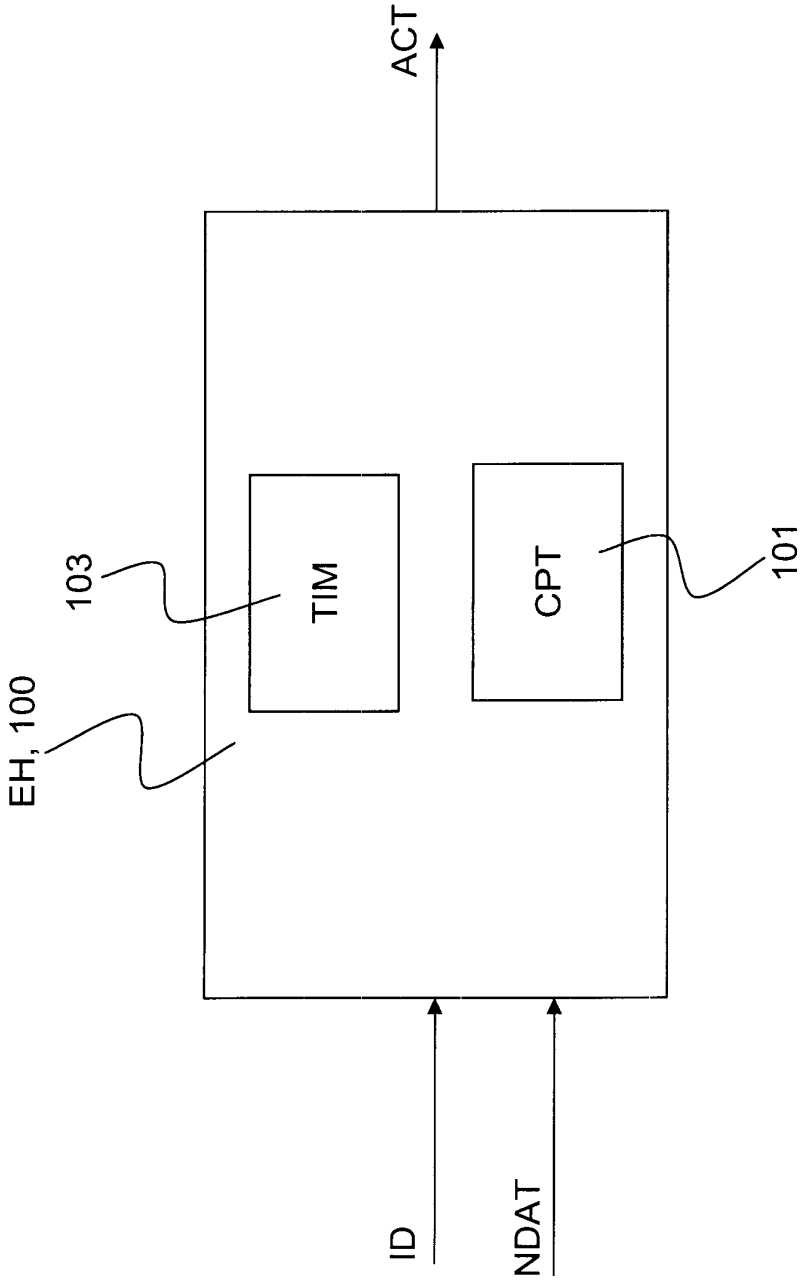


FIG.2

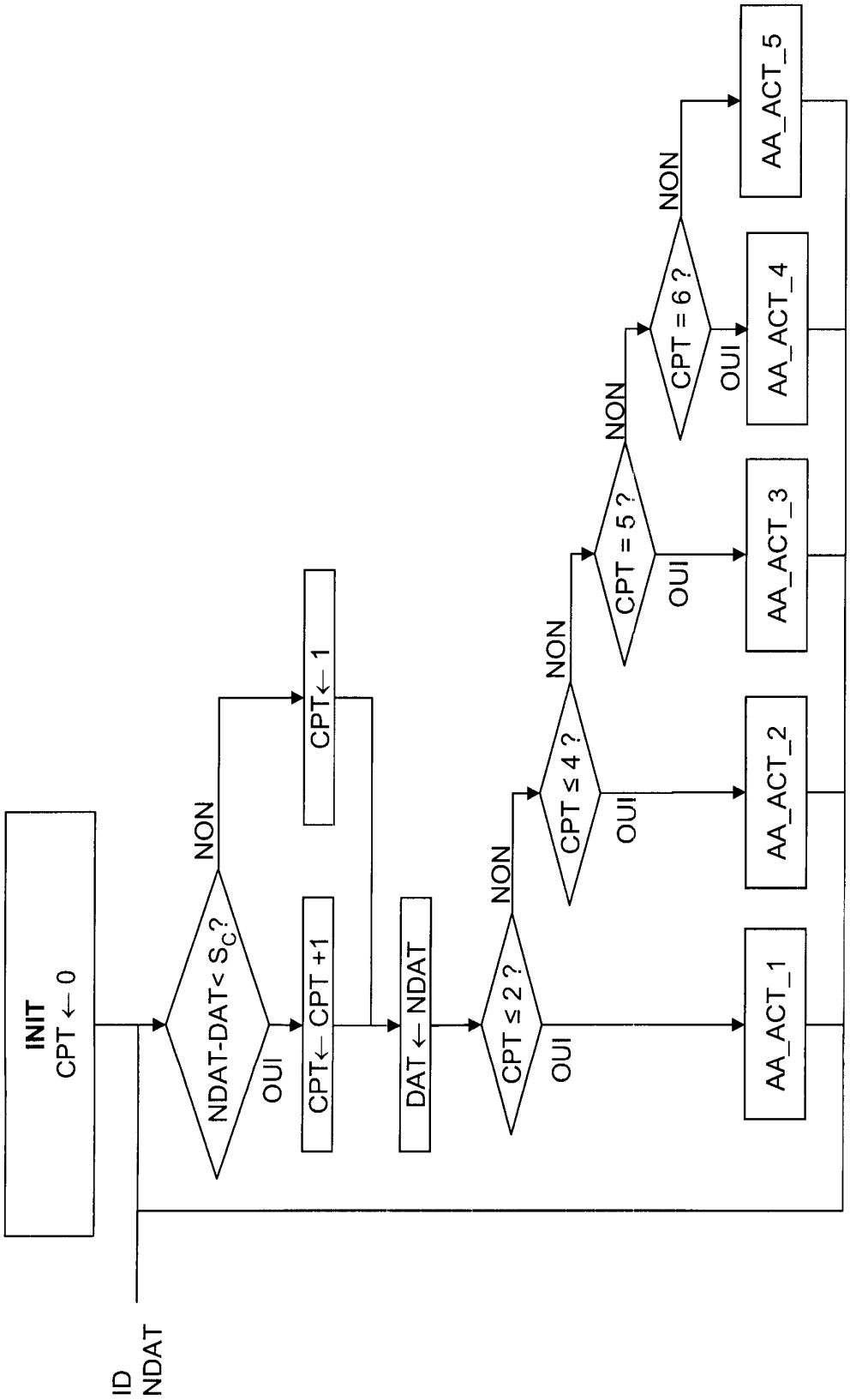

ID
NDAT
INIT
CPT ← 0
NDAT-DAT< S_C?
NON
OUI
CPT← 1
CPT← CPT +1
DAT ← NDAT
CPT ≤ 2 ?
NON
OUI
AA_ACT_1
CPT ≤ 4 ?
NON
OUI
AA_ACT_2
CPT = 5 ?
NON
OUI
AA_ACT_3
CPT = 6 ?
NON
OUI
AA_ACT_4
AA_ACT_5

FIG.3

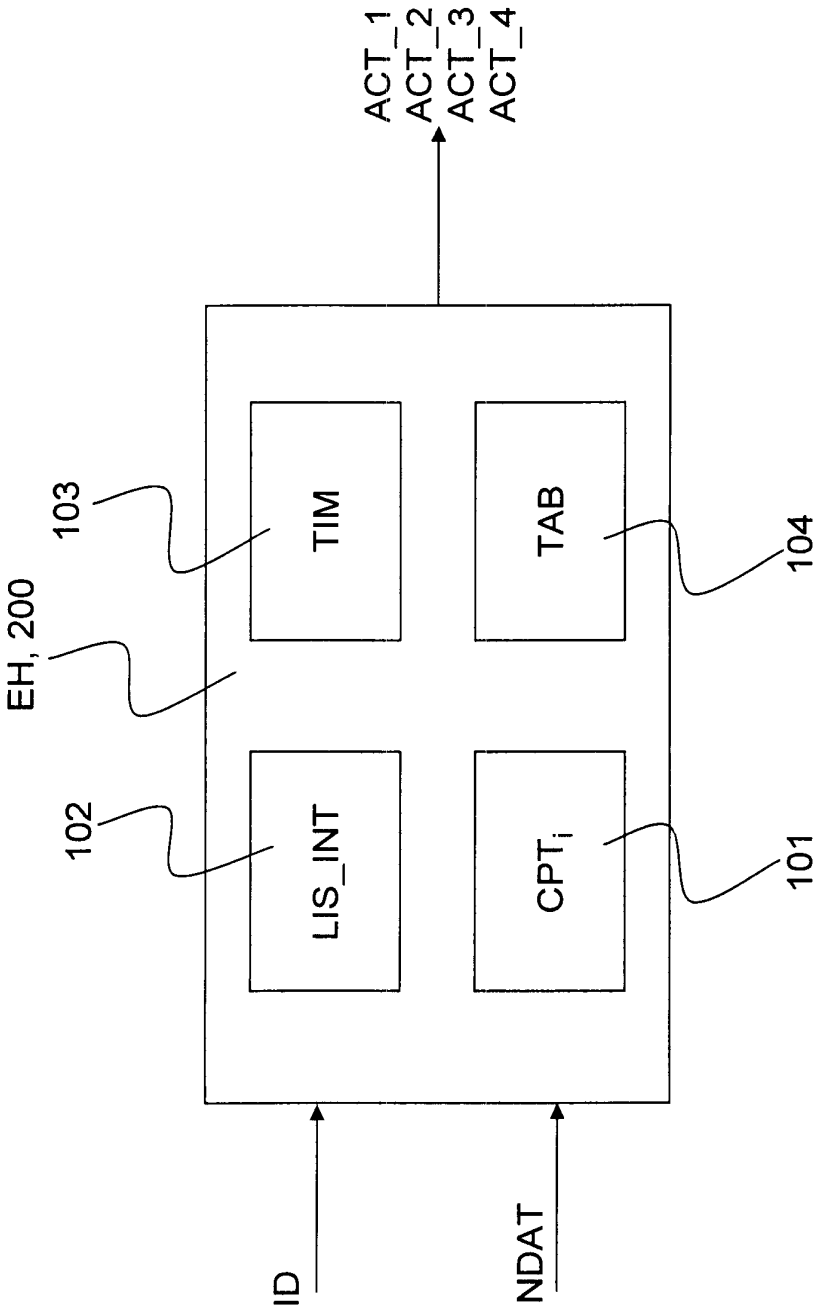

ACT_1
ACT_2
ACT_3
ACT_4
EH, 200
103
TIM
TAB
104
102
LIS_INT
CPT$_i$
101
ID
NDAT

FIG.4

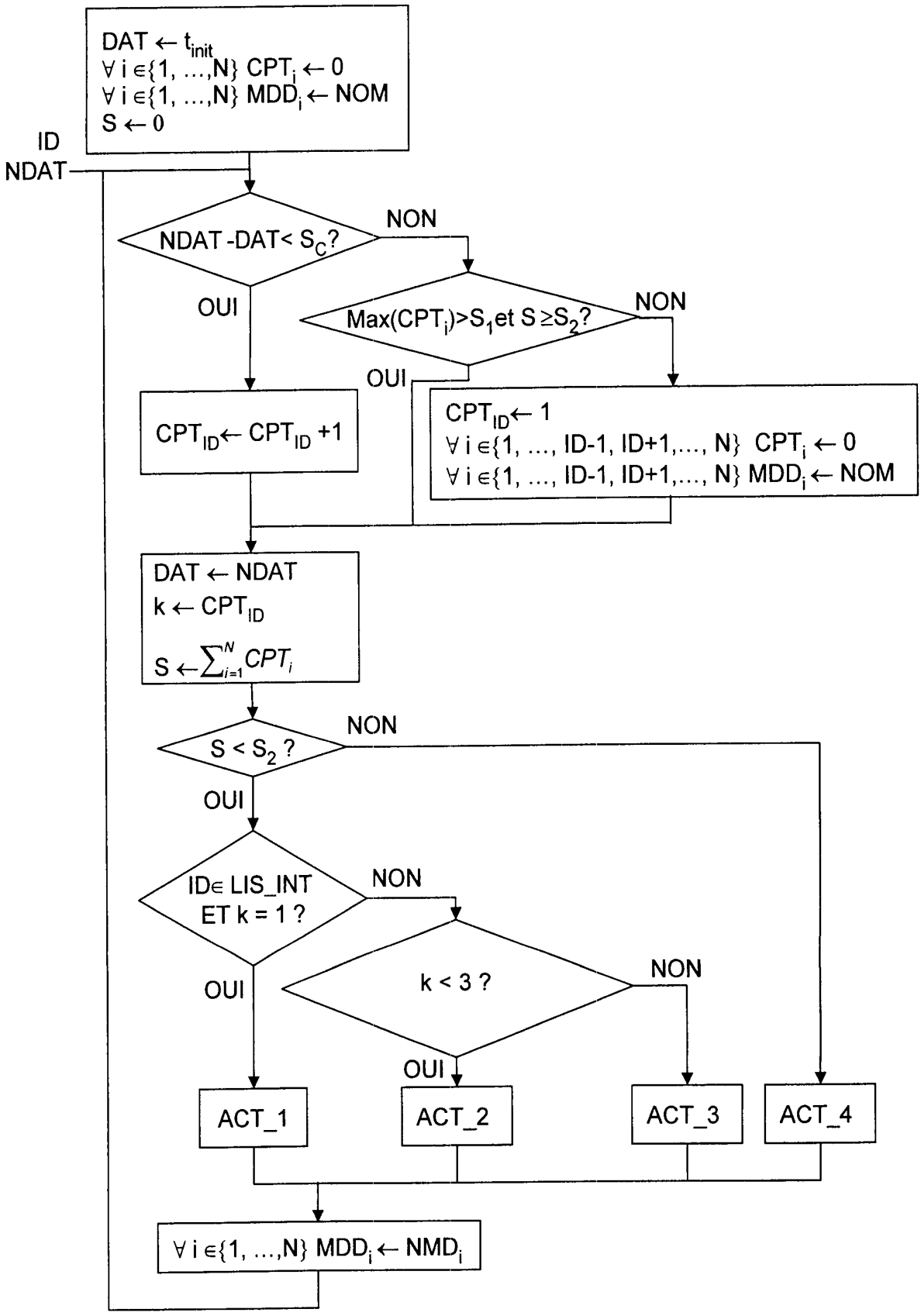

DAT ← t_{init}
∀ i ∈{1, ...,N} CPT_i ← 0
∀ i ∈{1, ...,N} MDD_i ← NOM
S ← 0
ID
NDAT
NDAT -DAT< S_C?
NON
OUI
Max(CPT_i)>S_1et S ≥S_2?
NON
OUI
CPT_{ID}← CPT_{ID} +1
CPT_{ID}← 1
∀ i ∈{1, ..., ID-1, ID+1,..., N} CPT_i ← 0
∀ i ∈{1, ..., ID-1, ID+1,..., N} MDD_i ← NOM
DAT ← NDAT
k ← CPT_{ID}
S ←$\sum_{i=1}^{N} CPT_i$
S < S_2 ?
NON
OUI
ID∈ LIS_INT
ET k = 1 ?
NON
OUI
k < 3 ?
NON
OUI
ACT_1
ACT_2
ACT_3
ACT_4
∀ i ∈{1, ...,N} MDD_i ← NMD_i

FIG.5

RÉPUBLIQUE FRANÇAISE

2910656

INSTITUT NATIONAL DE LA PROPRIÉTÉ INDUSTRIELLE

RAPPORT DE RECHERCHE PRÉLIMINAIRE

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

N° d'enregistrement national

FA 689878
FR 0611087

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		
A	US 2004/215997 A1 (ANDERSON GARY D [US] ET AL) 28 octobre 2004 (2004-10-28) * abrégé * * alinéas [0002], [0018], [0024], [0026] * * figures 1-6 * -----	1-22	G06F11/28 G06F9/455
A	US 6 453 430 B1 (SINGH DALJEET [US] ET AL) 17 septembre 2002 (2002-09-17) * abrégé * * colonne 8, ligne 49 - colonne 10, ligne 16 * * colonne 10, ligne 60 - ligne 65 * * colonne 12, ligne 14 - ligne 18 * * figures 1,2a,2b * -----	1-22	
A	US 2003/226056 A1 (YIP MICHAEL [US] ET AL) 4 décembre 2003 (2003-12-04) * abrégé * * alinéas [0030], [0036] * * figure 5 * -----	1-22	
A	US 5 305 455 A (ANSCHUETZ BRIGITTE D L [US] ET AL) 19 avril 1994 (1994-04-19) * abrégé * * colonne 3, ligne 38 - ligne 42 * * colonne 4, ligne 7 - ligne 53 * * figure 2 * -----	1-22	

DOMAINES TECHNIQUES RECHERCHÉS (IPC)

G06F

Date d'achèvement de la recherche	Examinateur
20 juillet 2007	Sabbah, Yaniv

CATÉGORIE DES DOCUMENTS CITÉS

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

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

EPO FORM 1503 12.99 (P04C14)

ANNEXE AU RAPPORT DE RECHERCHE PRÉLIMINAIRE RELATIF A LA DEMANDE DE BREVET FRANÇAIS NO. FR 0611087 FA 689878

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 20-07-2007
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
US 2004215997 A1	28-10-2004	JP 2004326764 A	18-11-2004
US 6453430 B1	17-09-2002	AUCUN	
US 2003226056 A1	04-12-2003	US 7017082 B1	21-03-2006
US 5305455 A	19-04-1994	AUCUN	

EPO FORM P0465