FACTA UNIVERSITATIS (NIŠ)
SERIES: ELECTRONICS AND ENERGETICS VOL. 6, NO. 1, (1993), 89-102

SURETE DE FONCTIONNEMENT DES SYSTEMES INFORMATISES AVEC L'APTITUDE DE TOLERANCE AUX FAUTES

Svetislav B. Kostić et Gilles Baratte

Résumé: Une approche correcte de la conception des systèmes informatisés doit respecter les objectifs de la sûreté de fonctionnement, comme la qualité de service que le système délivre, tout en acceptant que ces caractéristiques doivent être incorporées dans la structure de systéme et être validées. La sûreté de fonctionnement n'est ici considérée que par la fiabilité, la disponibilité et la sécurité, et par la tolérance aux fautes comme l'aptitude du système de survivre dans le cas d'apparition des fautes. Après de présentation d'une taxonomie des notions de base et d'analyse d'efficacité de la tolérance des fautes matérielles et logicielles, on présente les désavantages et surtout les limites de cette technique. Puis, comme conclusion, on offre comme l'objet de recherche à l'avenir, une nouvelle approache systèmique vers le concept des systèmes informatisés sûr et disponible, avec l'aptitude d'existence en présence des fautes.

1. Introduction

Au fil des années on a connu un élargissement considérable de la complexité des systèmes grâce à l'évolution importante et à la possibilité du mixage des technologies. A cause de cela les performances des systèmes sont accrues et devenues plus sophistiquées, et de plus en plus avec beaucoup de fonctions automatisées. Le concept général de ses systèmes est le système de technologies différents assisté par microprocesseur ou ordinateur, c'est-à-dire le système informatisé. La complexité des systèmes, la multitude des functions critiques ou non, les nouvelles technologies et surtout leurs mixages, tous ensemble sont élevés des risques de bon et de sûr fonctionnement

Manuscript received Juni 25, 1992.

S. Kostić is with l'Institut des Sciences Nucléaires de Vinča, Départment d'Electronique b.p. 522, 11001 Beograd, Yougoslavie. G. Baratte is with Centre d'Etude Nucléaires de Saclay, Départment d'Electronique et d'Instrumentation Nucléire F 91191 Gif-sur-Yvette Cedex, France.

du système, particulièrement dans des milieux hostiles. C'est pourquoi dans ces applications différentes on pose très souvent la question de la qualité de fonctionnement, mais avec la même gravité que la performabilité du système. Ceci a donné la notion de la sûreté de fonctionnement comme la qualité de service que le système délivre, qualité telle que les utilisateurs du systèm puissent lui accorder une confiance justifiée, [12]. Mais, il fault souligner que jusqu'à présent, dans la communauté des concepteurs et des utilisateurs, un système était habituellement consideré avec l'accent étant mis seulement sur son comportement extérieurement visible, c'est-à- dire sur sa fonction.

Les systèmes informatisés avec l'apptitude de tolérance aux fautes se trouvent aujourd'hui dans différentes applications et il faut s'attendre à ce que le nombre de ces systèmes croisse à l'avenir. La raison pour cela est d'abord dans le besoin de les appliquer et, de plus, dans les grandes possibilités de nouvelle technologie qui peut faciliter ces applications. La tolérance aux fautes est aujourd'hui mieux conçue et surout la possibilité de l'appliquer non seulement dans les systèmes spécifiques (systèmes avionic et spatiaux), mais aussi à l'industrie, aux opérations de banque et de commerce, aux affaires publiques, aux équipements médicaux, dans l'armement, etc.

Ici, dans ce papier, l'accent se pose plutôt sur la méthode de tolérance aux fautes comme une approche possible de la furniture la sûreté de fonctionnement. Mais il faut souligner que le mythe associé avec la technique de tolérance aux fautes conduit les non initiés de concluire qu'elle assure toujours des solutions plus fiables. Au contraire, l'application de cette technique sans précautions nécessaires conduit souvent à une diminution non percue de la fiabilité et de la disponibilité du systéme, c'est-à-dire on arrive à un effet contraire mais beaucoup plus cher. Cela signifie qu'il faut appliquer une approche systèmique de la conception du système sûr de fonctionnement et disponible, avec l'optimisation la performabilité et les objectifs de sûreté de fonctionnement par rapport au coût de cycle de vie. En méme temps, il faut essayer de surmonter les désavanteages et surtout les limites de cette technique et le vecteur d'avenir de la recherche devrait être vers le nouveau concept des systèmes informatisés.

2. Systeme et Surete de Fonctioment

Le système est défini par son comportement extérieurement visible, la performabilité, et par sa structure. Dans l'étude de la qualité de son fonctionnement, il doit être considéré comme une entité complexe, avec des processus et des interactions internes et avec des interactions différentes avec l'environnement. En même temps, il faut accepter que le système a sa

vie, sa durée, son coût total, et surtout des imperfections différentes qui se manifestent par l'apparition des défaillances. Au sens général et classique dans le système informatisé on distingue les composants de types matériel et logiciel. Les systèmes et les composants sont d'après leurs natures imparfaits, ce qui se manifeste par l'apparition de défaillances du système. La défaillance est survenue à cause de l'existence d'erreur, comme la partie de l'état d'un système qui est susceptible de provoquer une défaillance. La cause phénoménologique d'une erreur est une faute. Une erreur est la manifestation d'une ou plusieurs fautes dans le système, et une défaillance la manifestation d'une ou plusieurs erreurs sur le service, [12].

A cause de la défaillance, la vie du système est percue par ses utilisateurs comme une alternance entre deux états du service délivré, par rapport au service spécifié, le service approprié et le service inapproprié, [5]. La quantification de cette alternance conduit, en général, aux mesures de bases de la sûreté de fonctionnement: fiabilité, maintenabilité, disponibilité, et sécurité. Les mesures de base sont la fiabilité et la disponibilité, parce que la maintenabilité est moins utilisée. La sécurité est introduite à condition d'avoir la possibilité de mesurer les conséquences de la défaillance sur l'environnement du système. Pour améliorer les caractéristiques précédentes du système on peut passer par l'utilisation combinée d'un ensemble de méthodes de la sûreté de fonctionnement qui peuvent être classées de la façon suivante:

méthodes de la furniture de la sûreté de fonctionnement

-évitement des fautes, où on fait minimisation, par construction, de la possibilité d'apparition des fautes;
-tolérance aux fautes, où on fait délivrance, par redondance et par diversification de conception, d'un service conforme au service spécifié malgré l'apparition (passée ou présente) des fautes.

méthodes de la validation de la sûreté de fonctionnement

-suppression des erreurs, où on fait minimisation, par **vérification**, de la présence des fautes;
-prévision des erreurs, où on fait estimation, par **évaluation**, de la création, de la présence et des conséquences des erreurs, [5], [17].

Les définitions données pour les méthodes précédentes sont en effet des objectifs qui ne peuvent être complètement atteints. C'est pourquoi, il est nécessaire que les méthodes et les outils correspondants soient employés de manière combinée respectant les principes du génie sûreté de fonctionnement.

3. Methodes et Techniques de Tolerance aux Fautes

La tolérance aux fautes est basée sur le traitement des erreurs et des fautes. Cela signifie qu'on a accepté que les systèmes informatisés sont imparfait et qui doivent accomplir des traitements non seulement les entrées fonctionelles, mais également les erreurs et les fautes. De plus, c'est bien evident que toutes les erreurs et les fautes ne puissent pas être tolérées, c'est pourquoi l'étude profonde de la pathologie du système représente la physique du problème. Les sources des fautes se trouvent dans les entités de matériel et de logiciel, et aussi, trés souvent, elles sont dues à l'intervention incovenable d'un opérateur ou d'un réparateur du système. C'est la raison qu'on trouve habituellement un classement classique qui comporte troi grandes familles de fautes: fautes du matériel, fautes du logiciel, et fautes humaines. Pourtant, une considération plus générale permet un classement différent, où toutes les fautes dans les système informatisé peuvent être classées d'après: l'origine, de persistance temporelle et la possibilité d'élimination des fautes.Les fautes du système selon leur origine sont les suivantes:

fautes physiques

phénomènes physiques adverses, intérieurs (désordre physico-chimiques), ou extérieurs (perturbations dans l'environnement du système);

fautes humaines

imperfections humaines qui peuvent être:

.des fautes de conception, commises au cours de la conception (au sens large du terme, depuis les spécifications initiales jusqu'à la réalisation);

.des fautes opérationnelles, commises au cours de l'établissement des procédures d'opération, ou de maintenance;

.des fautes d'interaction, violations (par inadvertance ou de facon délibérée) des procédures d'exploitation ou maintenece, [5], [9].

Les fautes du logiciel sont les fautes de conception et les fautes du matériel sont plutôt les fautes physiques. Les fautes très inconvenentes au système redondant sont les fautes communes qui peuvent complètement annuler l'aptitude de tolérance aux fautes. Ce sont les fautes coïncidentes dans les voies redondantes du système, mais créées pendant la conception du système (au sens large du terme) ou pendant l'exploitation du systéme dues au mode commun. D'autre part, une étude des défaillance du système, du point de vue des conséquences sur l'environnement, donne la possibilité de classer les

fautes par gravité de conséquences, [12]. Plusieurs modes de défaillance peuvent être généralement considérés, ordonnés selon leur criticitè et regroupés en deux classes de criticité très différenciées:

-défaillances bénignes, où les interruptions résultantes ayant des conséquences qui sont comparable (généralement en termes de coût) à celles résultant d'accomplissement du service;
-défaillances malignes ou catastrophiques, où les interruptions résultantes ont des conséquences incommensurables avec celles résutantes d'accomplissement du service.

Le traitement d'erreur est destiné à éliminer l'existence d'erreurs, si possible avant qu'une défaillance ne survienne, qui de ce fait par des actions de recouvrement ou de compensation, [1], [16], [17]. Deux procédures sont habituellement accompagnées par la détection d'erreur. Le recouvrement d'erreur représente une procédure pour substituer un état erroné par un état exempt d'erreur, qui se fait de deux maniéres possibles: par la reprise d'erreur, où le système est ramené dans un état survenu avant d'erreur, et/ou par la poursuite d'erreur, où on passe à un nouvel état exempt d'erreur. Dans la compensation d'erreur (un masquage d'erreur) un état erroné comporte suffisamment de redondance pour permettre la délivrence d'un service approprié, où la redondance peut être: redondance du matériel, des informations, du logiciel, et en temps. L'application systèmatique de la compensation assure que toute erreur effective a été ramenée à l'état latent. Cependant, ceci peut par là même donner lieu à une diminution non perçue de la redondance. La détection d'erreur représente l'action de déceler l'existence d'erreur appliquée lorsque le recouvrement d'erreur est utilisé.

Le traitement de fautes est déstiné à éviter qu'une faute ne soit activée à nouveau, ce qui se fait par réparation ou par reconfiguration, [1], [12], [17]. La réparation représente la partie des activités de la maintenance corrective destiné à supprimer des fautes qui ont été effectives. La réparation comporte les opérations suivantes: la diagnostic de faute, où une action est effectuée de déterminer les causes des erreurs en termes de localisation et de nature; la passivation de faute, où une action est effectiée pour empêcher une nouvelle activation de faute, qui est accomplie en retirant les composants considérés comme fautifs du processus d'exécution ultérieur. Une action très importante est la reconfiguration du système, où un changement de structure du système est effectué par la passivation de composant défaillant afin que le système soit capable de délivrer le même service qu'auparavant ou un service dégradé.

Les techniques de tolérance aux fautes sont nombreuses, mais parmi elles

on peut définir deux techniques élémentaires:

-détection-recouvrement d'erreur,
-masquage d'erreur.

Ces techniques se combinent de manières différentes, avec l'application ou non des critères de diversification dans la réalisation d'éléments redondants, variants des composants ou sous-systèmes, ou de système même. Elles sont basées sur le traitement d'erreur et de faute, au sens d'un traitement ou plusieurs, mais consécutifs. Mais s'il y a des traitements multiples, par voies multiples, on parle d'une autre technique, largement utilisée, la technique de la diversification fonctionnelle, ou la redundance en fonction. On applique cette technique même pour la tolérance aux fautes physique que pour la tolérance aux fautes de conception. Les voies multiples représentent les variantes ou versions du système qui doivent délivrer des services identiques. Un moyen de décision existe, qui doit choisir, parmi les réponces des variantes différentes, la réponse exempt d'erreur. D'autre part, les variantes du système peuvent être identiques ou différentes selon le critére d'indenpendance ou non des défaillances des variantes. Dans le cas de défaillances indépendantes, les variantes peuvent être identiques, ce qui est le cas du matèriel (les fautes physiques seulement). Au contraire, les variantes doivent être différentes, ce qui est le cas du logiciel (les fautes de conception sont dominantes). Si les variantes sont différentes on parle de la diversification de conception réalisée de manière différentes du point de vue de: conceptions des voies, concepteurs des voies, technologies des voies, c'est-à-dire des critères de diversification.

4. Tolerance des Fautes Materielles

Les fautes matérielles sont plutôt des fautes physique, mais très souvent les fautes do conception du matériel ne sont pas négligeables. La tolérance des fautes matérielles est accomplie par des techniques de base: détection-recouvrement et masquage d'erreur. C'est un regroupement grossier parce que en réalité les techniques sont toujours combinées avec l'application ou non de diversification fonctionnelle et/ou de diversification en conception. La redondance, selon le critère de "fonctionnement ou non des composants", peut être passive ou active, ou parfois la combinaison des deux. D'autre part, regardant l'architecture du systéme redondant, d'aprés le critére "d'apptitude de reconfiguration du système", on distingue deux grandes classes des systèmes: systèmes en redondance statique ou dynamique, [8], [9], [11], [17].

La redondance statique permet le masquage d'erreur du composant en ajoutant plusieurs copies de même composant en parallèle mais dans une

configuration du systéme fixe. Toutes copies redondantes sont habituellement identiques et en marche d'œuvre permanente (active). La multitude de copies présente que la redondance statique est plutôt massive, avec de la réparation des composants redondants (copies) défaillants habituellement hors ligne. Les réalisations techniques des redondances statiques sont différentes, nombrées et assez bien connues, comportant deux techniques bases appliquées depuis longtemps dans des versions pures ou combinées. Ces techniques sont les suivantes:

-redondance en matériel, (N modules en redondance), surtout la redondance en trois, et la redondance "k/N" largement utilisée, spécialement 2/3 et 2/4, [7];
-redondance en information, (code correcteur).

La redondance dynamique englobe la possibilité de la reconfiguration du système comme la réponse à l'apparition des fautes. Cela assure le fonctionnement continu du système, mais quelquefois d'une manière dégradée. La reconfiguration de sa part est activée dès que la faute est détectée et le succès d'exécution de la reconfiguration est déterminée par l'apptitude de détection qu'elle englobe la confirmation d'erreur. La reconfiguration est suivie par recouvrement d'erreur et par le traitement des fautes qui est appliqué par le diagnostic et par la passivation, c'est-à-dire, par réparation manuelle (hors ligne) ou automatique (en ligne). A cause de la possibilité de reconfiguration du système et, aussi, par l'application plus évoluée de la détection et du recouvrement d'erreur, la redondance dynamique est moins massive et plutôt sélective par comparaison à la redondance statique. Les techniques de la redondance dynamique sont nombreuses et différentes, et une taxonomie claire est difficile à poser. Pour accentuer la souplesse du systéme au point de vue du recouvrement d'erreur et de reconfiguration du système on peut distinguer deux classes:

-techniques à redondance massive,

où le recouvrement et la reconfiguration sont faibles, (système duplex; NMR reconfigurée - redondance hybride et redondance avec l'arbitre adaptif, 5MR configuration reconfigurée);

-techniques à redondance souple,

où le recouvrement et la reconfiguration plus évolués sont appliqués.(redondance par réserve, par un traitement réduit et par triage).

Les conclusions générales sur la tolérance les fautes matérielles sont les suivantes:

-toutes les structures redondantes peuvent être classées en deux grands groupes selon la conception de la structure redondante qui assure, ou la performabilité de structure sous risque d'avoir l'interruption de délivrance de service, ou la continuité de marche du système au prix de dégradation des performances du systéme;

-les structures redondantes possibles sont nombreuses ce qui signifie qu'il n'y pas une méthode propre qui assure la solution unique de conception du systéme;

-les redondances exposées existent depuis longetemps, ce qui signifie qu'on ne peut pas s'attendre à voir une autre structure nouvelle, mais plutôt une nouvelle combinaison d'existantes surtout en conbinaison avec des avantages et des possibilités de techniques et de technologies modernes;

- la mauvaise caractéristique de tous est la sensibilité sur les fautes communes dont l'apparition provoque la perte de l'apptitude de tolérance aux fautes;

-la redondance peut s'appliquer à tous les niveaux du système, (sous-systèmes, ... , composants), et même au niveau des systèmes. C'est pourquoi, la redondance est en liason forte avec de la fédération du système;

-jusqu'à ce jour, on utilise la redondance d'une manière intuitive, à la base du retour d'expériences et avec l'analyse plutôt de la fiabilité de la structure redondante. Il n'y a pas ni conception ni analyse complètes comportant des objectifs de sûreté de fonctionnement et surtout du coût de cycle de vie.

5. Tolerance des Fautes Logicielles

Les fautes logicielles sont les fautes de conception qu'elles restent dans le logiciel si n'importe quelles techniques, d'évitement des fautes et/ou de supression des fautes, sont appliquées. C'est pourquoi, il est indispensable d'appliquer la technique de tolérance aux fautes, où la nature des fautes logicielles, qu'elles soient selon leur origine, conditionnent d'une part le masquage d'erreur et, d'autre part, la diversification de conception.

Le masquage d'erreur de logiciel signifie l'assurance de la redondance du logiciel, ce qui se fait par des exécutions multiples du logiciel. Les exécutions multiples, d'une ou de plusieurs variantes du logiciel présente la méthode de base de la tolérance aux fautes logicielles. Les réplications d'exécution du logiciel sont, en général, d'ordre N ($N \geq 2$), mais selon la manière de réalisation on peut distinguer:

-réplication en temps, par reprise, surtout dans une variante unique du logiciel;

-réplication par plusieurs variantes du logiciel, où l'exécution multiple est parallèle ou séquentielle.

La diversification de conception présente une approche du développement des variantes de même système selon les critères de diversifications différentes. Les critères de diversification peuvent se regrouper dans les classes suivantes:

-diversification de méthodes et de techniques de développement, de validation et de réalisation du logiciel;

-diversification d'architecture et d'ensemble du matériel indispensable dans l'exécution du logiciel, réalisé avec une ou plusieurs voies, ou dans un sens combiné;

-diversification de conception du point de vue des équipes qui réalisent des variantes du logiciel, avec la diversification en éducation, formation et expérience, de différentes localisations géographiques, de différentes origines ethnique;

-diversification de technologie du matériel, de langage appliqué, des compilateurs, des données utilisées, des algorithmes;

-diversification de spécifications du point de vue des équipes qui définissent des spécifications, [4].

Pour améliorer l'efficacité d'applications des techniques précédentes il faut faire la fédération du système mettant en œuvre une ou plisieurs sous-functions du système. La fédération du système est une notion très importante dans le génie de sûreté de fonctionnement et les critères pour la fédération sont différentes. Un des critères est le confinement de la défaillance de tout système, où la défaillance ne devant pas empêcher la réalisation de la fonction globale du système, éventuellement dans un mode dégradé, [2]. Il faut seulement noter que l'approche de fédération conduit généralement à un grand nombre de sous-systèmes ou composants, supérieur à ce qui serait nécessaire en termes de traitement. Par exemple, le système de commande de vol de Bœing 756/767 ne comporte pas moins de 80 microprocesseurs d'un point de vue fonctionnel et 300 en tenant compte des redondances, [5].

Les techniques pratiques et de base de la tolérance les fautes logicielles sont les suivantes:

-block de recouvrement,
(recovery block-RB), [1];

-programmation N-version,
(N version programming-NVP), [3], [4], [13];

-programmation N-version avec auto-contrôle,
(N self-checking programming-NSCP), [6].

Dans les trois techniques il est possible d'apercevoir la structure qui ressemble à la redondance du matériel, mais avec des structures simples. La redondance à N variantes, en général, c'est la redondance statique. La redondance de RB est la redondance dynamique passive, et NVP et NSCP sont les redondances dynamiques actives. Normalement, on peut espérer voir un autre type de masquage en logiciel mais qui est déjà vue en matériel. Les exigences essentielles de tolérance aux fautes sont la diversification de conception et l'indépendance des variantes qui ne sont pas pour le présent assez précises et clarement définies, [14], [15]. Les systèmes réels où on trouve la diversification de conception sont nombreux et différents, ainci que les systèmes de sécurité, les systèmes commerciaux, et les systèmes dans l'industrie, [2], [9], [13], [18].

6. Objectifs de la Surete de Fonctionnement

Une approche correcte de la conception des systèmes doit respecter les objectifs de la sûreté de fonctionnement, tout en acceptant que ces caractéristiques doivent être incorporées dans la structure de système et être validées. Les objectifs de la sûreté de fonctionnement ne sont ici considérés que par la fiabilité, la disponibilité et la sécurité, et par la tolérance aux fautes comme l'aptitude du système de survivre dans le cas d'apparition des fautes. Dans l'application des systèmes informatisés, on mélange très souvent ces notions, surtout la fiabilité et la sécurité. On impose souvent des exigences sur la fiabilité en espérant avoir également un résultat acceptable sur la sécurité. Il est très rare que l'on impose des exigence bien définies séparment sur la fiabilité et la sécurité. Il faut aussi noter qu'un système assez fiable ne signifie pas qu'il est disponible, et qu'une assez bonne disponibilité ne signifie pas que la sécurité est aussi assurée. D'autre part, le mythe associé avec la technique de tolérance aux fautes conduit les non initiés de concluire qu'elle assure toujours des solutions plus fiables. De plus, l'application de cette technique sans précautions nécessaires conduit souvent à une diminution non perçue de la fiabilité et de la disponibilité du systéme, c'est-à-dire on arrive à un effet contraire mais beaucoup plus cher.

Les trois caractéristiques précédentes doivent être considérées par la construction et par la validation. Le problème de la construction est liée aux procédures de conception du système, ce qui n'est pas l'objet de cette papier parce que demande une considération séparément, [9], [10]. Pourtant, il faut voir pourquoi dans la mise en œuvre pratique il n'y a pas encore des valida-

tions satisfaisantes. La cause se trouve-t-elle dans les méthodes de validation mêmes ou chez les concepteurs des systèmes, qui ne les appliquent pas du tout ou de manière incorrecte? En principe, il est sûr que les concepteurs ont une part de responsabilité important mais aussi tous les autres intervenants dans la réalisation du projet (utilisateur, fabricant). Les cas de validations correctes sont rares et quand on les utilise, c'est une approche formelle et non complète, avec comme objectif principal la satisfaction d'exigences aussi formelles. Lorsque l'utilisateur ne connait pas assez bien le probléme, il accepte les résultats du concepteur sans vérifications sérieuses. C'est le cas habituel, tout les monde est satisfait et personne n'est visiblement pénalisé.

L'état de fait précédent se retrouve partiellement aussi dans les méthodologies mêmes et dans les possibilités de les utiliser facilement pour le matériel, le logiciel et le système complet. D'autre part, l'évaluation de la sûreté de fonctionnement du matériel est assez bien définie mais pas toujours facile à réaliser. Pour les logiciels, le problème est plus complexe et c'est la raison d'existence d'un grand nombre de modèles mathématiques pour la fiabilité, où chaque modèle remporte de succs limités. Pourtant, en conclusion de cette partie assez pessimiste, il faut constater qu'il y a aussi des solutions possibles. Le chemin proposé est l'assurance d'un ensemble complet de méthodologies quantitatives et qualitatives, plutôt sous forme d'outils logiciels, vérifiés et prouvés, et correctement utilisés respectant des règles de processus de conception du système. L'approche doit être vers le système considéré comme un tout suivant la croissance des objectifs de la sûreté de fonctionnement.

L'aptitude de tolérance aux fautes ou plutôt le principe de survivance dans le cas d'apparition des fautes, représente un principe important surtout pour les systèmes informatisés. Les techniques de réalisation de ce principe ne sont ni nombreuses, ni suffisamment efficases. Ces techniques sont la tolérance aux fautes, la défence en profondeur, [9], et partiellement la diversification de conception au sens gènèral. Parmi ces trois techniques, la tolérance aux fautes est plus accentuée et habituellement on attend beaucoup d'elle. Dans le matériel, où les fautes sont les fautes physiques, la tolérance aux faute remporte un certain succès. Dans le logiciel, où les fautes sont plutôt des fautes de conception, la tolérance aux fautes remporte un succès plus modest, même avec un grand effort d'application de diversification par des critères différentes. De plus, il faut aussi souligner que la tolérance des fautes matérielles et logicielles comporte seulement un petit nombre des fautes anticipées, indépendantes et non simultanées. En tout cas, cette technique n'est pas suffisement efficace et elle est sans perspective dans les systémes qui sont de plus en plus informatisés.

7. Conclusion

Toutes les méthodes et techniques précédentes de l'assurance de sûreté de fonctionnement des systèmes informatisés, qui peuvent donner des solutions satisfaisantes pour des systèmes simples, ne représent pas des méthodes et techniques de l'avenir. Elles sont plutôt applicables au matériel, mais dans un milieu où le logiciel est dominat, et surtout pour le système considéré comme un tout, il faut chercher ou attendre des méthodes ou techniques nouvelles complètement différentes. C'est pourquoi, une autre approche possible et payante à l'avenir se trouve dans le domaine de recherche où il faut essayer de surmonter les désavantages et surtout les limites des méthodes précédentes, mais respectant la réalité. D'abord, cette réalité est le système informatisé de l'avenir, automatique avec plus d'intelligence par comparaison aux systèmes d'aujourd'hui, mais avec la grande importance du logiciel. Il faut s'attendre que les méthodes d'évitement et de suppression des fautes ne soient pas complètement efficaces, particulièrement pour les fautes non anticipées. Actuellement un système sans fautes est une idéalisation vers laquelle on se dirige obstinément en dépensant beaucoup d'efforts mais cela sans succès complet. Cela signifie qu'il faut accepter les fautes comme la réalité, c'est-à-dire accepter que les systèmes sont imparfaits.

Dans les systèmes imparfaits il est important, au lieu du principe de survivance, incorporer le principe d'existence avec des fautes. L'aptitude de survivance, réalisée par les méthodes de la tolérance des fautes, dépend de la redondance prévue quelle ne couvre q'un petit nombre des fautes contrôlées et, c'est pourquoi, cette aptitude pourrait être dépenser. La notion d'existence se réfère à l'aptitude de vivre en présence des fautes, quelle doit être réaliser par la construction de ce principe dans la structure fonctionnelle du système, comportant l'architecture matérielle et les fonctions sophistiquées du logiciel, [9]. En ce moment, le problème technique de réalisation de ce principe est difficile à resoudre, mais profitant des possibilités de technologies contemporains et surtout de leur évolution rapide, il existe l'imperative de recherche des approches nouvelles et plus adequates. Le vecteur d'avenir de la recherche devrait être vers le concept des systèmes qui ressemblent aux systèmes biologiques, en tant qu'objectif difficile à atteindre, mais comme un modèle des systèmes d'avenir plus réel et plus acceptable. C'est pourquoi il faut avoir une approche avec un autre concept du système informatisé, profitant des caractéristiques plus sophistiquées des nouvelles technologies, comme les systèmes répartis, les réseaux de neurones, l'intelligence artificielle, les systèmes artificiels de neurones, ... , en fonction de leur évolution.

Les objectifs de conception des systèmes doivent être considérés par le

concept même du système et par le processus de conception. D'après des analyses précédentes ce sont deux domaines importants où il faut chercher des approches nouvelles comportant le principe d'existence avec des fautes. Cela doit être l'approche systèmique vers le concept du système dynamique, automatique et intelligent, avec l'aptitude d'existence en présence des fautes et avec la possibilité d'évolution. Dans le processus de conception du système le but principal qui se pose doit être une approche systèmique de la conception du système sûr de fonctionnement et disponible, avec l'optimisation la performabilité et les objectifs de sûreté de fonctionnement par rapport au coût de cycle de vie.

REFERENCES

1. T. Anderson and P. A. Lee: *Fault Tolerance - principles and practice.*. Prentice Hall, New York, 1981
2. T. Anderson (ed.): *Dependability of resilent computers.*. BSP Professional Books, London, 1989.
3. J.Arlat, K.Kanoun, and J. C. Laprie: *Dependability evaluation of software fault-tolerant computing.*, Tokyo, Japan, June 1988, pp.142-147.
4. A. Avizienis: *The N-version approach to fault-tolerant software.*, IEEE Trans.on Soft.Eng., SE-11, 12 (1985), pp.1491-1501.
5. A. Avizienis and J. C. Laprie: *Dependable computing - from concepts to design diversity.*, IEEE Proceeding, vol.74, 5 (1986), pp.629-638.
6. A. Cheilan and J. C. Laprie: *Software fault-tolerance: why, how, how much.*, Rapp.LAAS, No 87.077, Toulouse, France, 1987.
7. B. S. Dhilon and S. N. Rayapati: *Comparative reliability analysis of simplex and redundant systems.*, Microelectron.and Reliab., vol 25, 2 (1985), pp.343-356.
8. B. W. Johnson: *Design and analysis of fault-tolerant digital systems.*, Addison Wesley Pub.Comp., New York, 1989.
9. S. B. Kostić: *Sûreté de fonctionnement et tolérance aux fautes - systèmes critiques - systèmes de protection des rèacteurs nuclèaires.*, Rapp.DTA/LETI/DEIN/SAI-90-026, 1990.
10. S. B. Kostić, Z. Pendić and V. Arandjelović: *Concept of computer-based systems with embedded logistics support functions.*. Proceedings of 8th Int.Logistics Symp., ILS SOLE, Madrid, 1992.
11. P. E. Lala: *Fault-tolerant and fault-testable hardware design.*. Prentice Hall, New York, 1985.
12. J. C. Laprie: *Sûreté de fonctionnement des systèmes informatiques et tolerance aux fautes: concept de base.* Techniques et sciences informatiques, vol.4, 5 (1985), pp.419-429.
13. J. C. Laprie et al: *Hardware and software fault tolerance: definition and analysis of architextual solutions.*. Proceedings of 17th Int.Symp.on fault tolerant computing, FTCS'17, 1987, pp.116-121.
14. B. Littlewood and T. Anderson: *Reliability modeling for fault-tolerant software.*. in ref.18, 1988, pp.169-182.
15. B. Littlewood and D. R. Miller: *Conceptual modeling of coincident failures in multiversion software.*. IEEE Trans.on Soft.Eng., SE-15, 12 (1989), pp.1596-1614.

16. B. Randel, P. A. Lee and P. C. Treleaven: *Reliability issues in distributed computing systems.*. Proceedins of 5th Symp.on Reliab.in Distributed Software and Datebase Systems, Los Angeles, 1986, pp.123-165.

17. D. P. Siewiorek and R. S. Swarz: *Theory and practice of reliable system design.*. The Digital Press, Bedford Mass., 1982.

18. U. Voges (ed.): *Software diversity in computerised control systems.*. Springer-Verlag, Wien, 1988.