les notes scientifiques et techniques de l'i.n.r.s

Septembre 2002

224

GESTION DES FONCTIONS DE SECURITE PAR AUTOMATE PROGRAMMABLE DEDIE A LA SECURITE (APIdS)

DEI-SVALDI D., KNEPPERT M.

N° Edition : NS 0224

INRS

GESTION DES FONCTIONS DE SECURITE PAR AUTOMATE PROGRAMMABLE DEDIE A LA SECURITE (APIdS)

DEI-SVALDI D. , KNEPPERT M.

Département « Ingénierie des Equipements de Travail »

Résumé

La création ou la rénovation d'unité de fabrication automatisée conduit l'utilisateur à initier une réflexion quant au choix technico-économique à opérer pour développer son application. C'est dans ce cadre qu'il est nécessairement amené à s'interroger et prendre position sur le système de gestion des sécurités et notamment sur l'utilisation d'automates programmables.
L'objectif de cette publication est de guider l'utilisateur dans sa réflexion en présentant les étapes qu'il aura à franchir pour atteindre l'objectif qu'il s'est fixé notamment sur la sécurité des personnes exploitant l'équipement.
Dans une première phase, nous invitons le lecteur à prendre connaissance de la réglementation qui accompagne l'utilisation des APIdS.
Dans une deuxième phase, nous présentons les principales approches industrielles permettant, selon les exigences de sécurité requises et la nature de l'application, de choisir la structure du système de gestion des sécurités directes la mieux adaptée.
Dans une troisième phase, nous posons les principaux problèmes à résoudre lorsque la gestion des sécurités sera confiée à un APIdS.

Mots-clés : Automates programmables – Gestion des sécurités – Sécurités directes

SOMMAIRE

1 - INTRODUCTION

En 1984, l'INRS [1] (CND 1502-117-84) recommandait de ne pas faire confiance au seul Automate Programmable Industriel (API) pour assurer la gestion des fonctions de sécurité et il était proposé d'assurer celle-ci par une logique câblée extérieure à la commande gérée par l'API. Depuis certains fabricants proposent ou vont proposer de nouveaux API appelés Automate Programmable Industriel dédié Sécurité (APIdS) devant pouvoir assurer à eux seuls la gestion des fonctions de sécurité. Ce document aborde la problématique liée à l'exploitation et à la mise en œuvre des fonctions de sécurité sur les machines ou équipements pilotés par un APIdS. Dans un premier temps, nous rappellerons la position prise en 1998 par le Ministère de l'Emploi et de la Solidarité [2] (note relative à l'acceptation de certains automates programmables pour gérer des fonctions de sécurité sur machine). Nous citerons ensuite les différentes architectures permettant de gérer les fonctions de sécurité ainsi que les solutions existantes, nous analyserons les architectures internes des APIdS actuellement sur le marché et les problèmes de validation liés aux différents types d'applications rencontrées.

2 - ETAT DE LA REGLEMENTATION

En l'état de la technique, il est en toute rigueur impossible de s'assurer intégralement du respect de l'exigence essentielle 1.2.7 de l'annexe 1 au décret 92-767 du 29 juillet 1992 (transposant l'annexe 1 de la directive Machines 98/37/CEE) dans le cas d'utilisation d'automates programmables standards *(un défaut affectant la logique du circuit de commande ou une défaillance ou une détérioration du circuit de commande, ne doit pas créer de situations dangereuses)*.

C'est pourquoi, la note établie par le Ministère de l'Emploi et de la Solidarité [2], propose à l'ensemble des industriels et agents de prévention concernés de choisir le ou les types de technologie appropriés à l'analyse des risques effectuée en prenant en considération les précautions élémentaires suivantes :

- la gestion des fonctions de sécurité doit être séparée de la gestion de la partie fonctionnelle,
- les fonctions de sécurité doivent être figées et non modifiables par l'utilisateur.

Le respect de ces critères est fondamental. Il est en effet normal que l'exploitant puisse avoir accès au programme gérant son processus de fabrication. Par contre, ces modifications ne doivent en aucun cas dégrader le niveau de sécurité de l'équipement, ce qui est assuré si les deux conditions précédentes sont respectées.

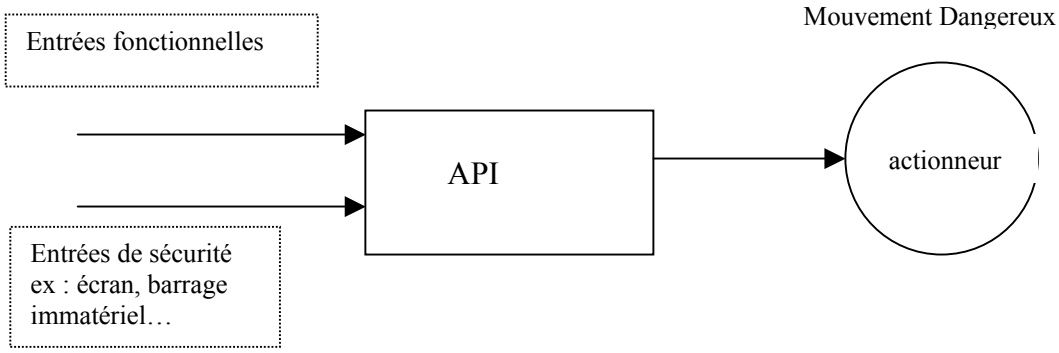
On comprendra aisément que l'exploitant ne puisse pas intervenir sur le niveau de sécurité de son installation sans s'entourer de précautions. En effet, il faut garder en mémoire que l'obtention d'un niveau de sécurité donné résulte d'une analyse de risques, d'un choix des dispositifs de sécurité les mieux adaptés, le cas échéant d'une concertation avec le personnel intervenant, et surtout de la validation de l'ensemble. De ce fait, la moindre modification, même partielle, requiert une nouvelle validation sans quoi elle pourrait avoir de graves conséquences sur la sécurité du personnel.

3 - ARCHITECTURE

Avant d'aborder les différentes architectures possibles, il y a lieu d'examiner la part prise par le circuit de commande dans la sécurité globale de la machine. En effet, si sur certaines machines présentant un niveau de risque très élevé, la conception du circuit de commande contribue de manière importante à la prévention des risques d'accidents (cas des presses, notamment), il arrive aussi que la sécurité repose pour l'essentiel non sur le circuit de commande mais sur d'autres moyens tels que la mise sous carter, l'éloignement, la mise en place de procédure d'intervention, etc. Dans ce contexte, les effets prévisibles d'une éventuelle défaillance du circuit de commande apparaissent comme négligeables dans l'appréciation des risques [3] (EN 292).

Les paragraphes suivants présentent les différentes architectures théoriques possibles pour la gestion des fonctions de sécurité sur une machine.

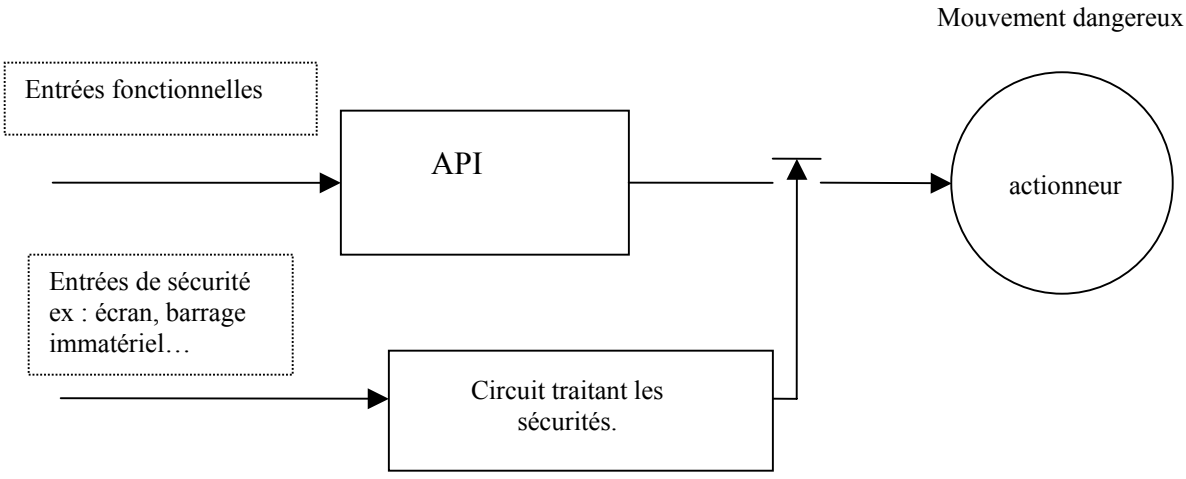
3.1. API gérant la commande et les sécurités



L'observation de ce synoptique montre qu'un mouvement dangereux peut se produire suite à une défaillance de l'API, les sécurités ne pouvant plus intervenir pour arrêter ce mouvement dangereux.

Ce comportement est dû au fait qu'un API standard n'a pas été conçu pour détecter toutes ses défaillances internes et adopter une position de repli en sécurité lorsque celles-ci se produisent. Pour ces raisons, l'utilisation d'un API standard n'est pas admise pour gérer les fonctions de sécurité directe sur une machine.

3.2. API gérant la commande, le circuit traitant les sécurités étant séparé

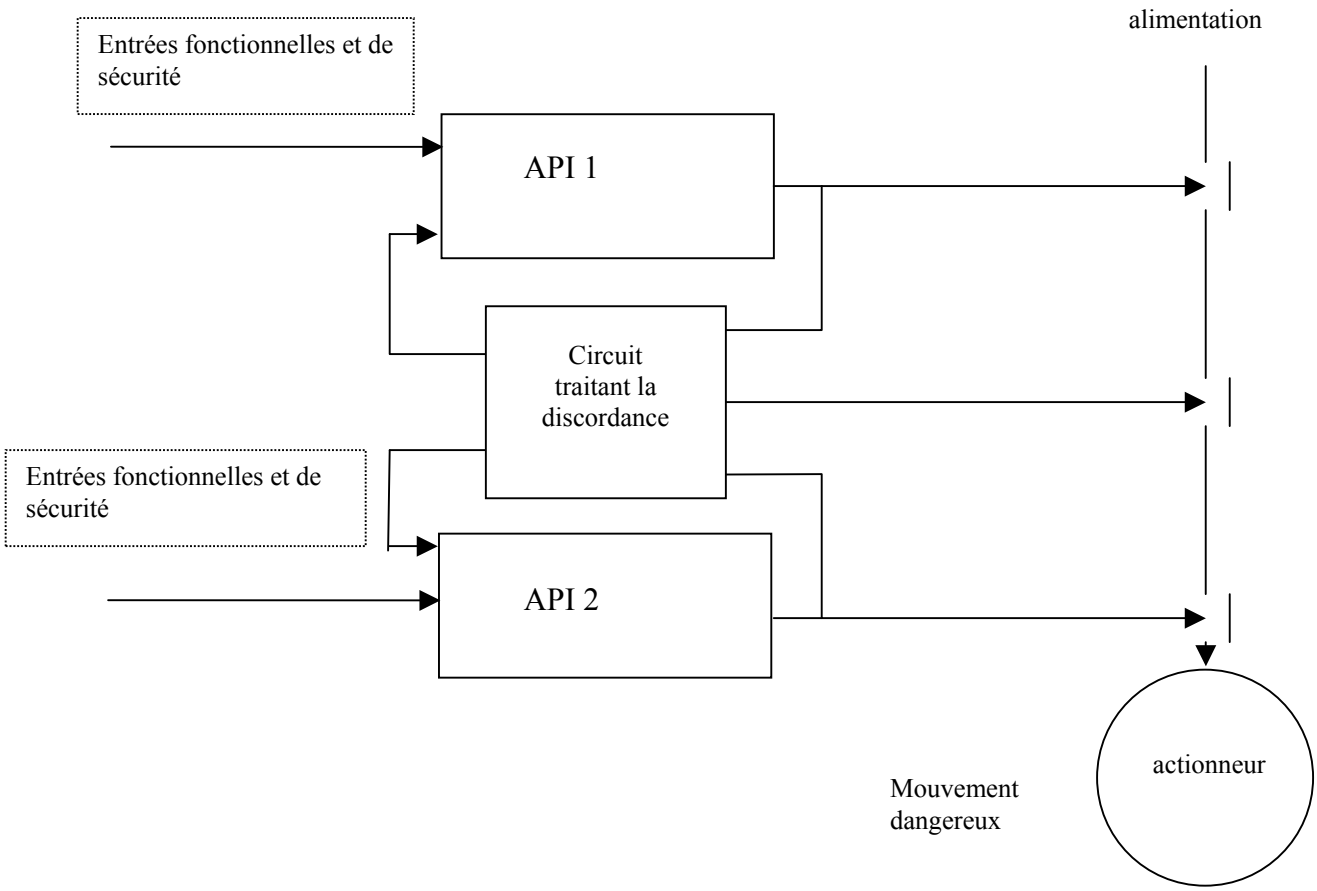


On constate qu'il n'existe pas de liaison directe entre une commande intempestive provenant de l'API et le mouvement dangereux. Le traitement des sécurités par un circuit spécifique validé permet la commande de la mise en sécurité de la machine même si la sortie de l'API commande un mouvement intempestivement. Cette architecture permet l'utilisation d'un automate car le traitement séparé des sécurités annihile les mouvements dangereux malgré la défaillance de l'API.

Cette solution se rencontre fréquemment et elle est recommandée lorsqu'elle peut être appliquée, car elle permet de valider aisément la sécurité d'un système global, complexe ou non, en validant seulement le circuit traitant les sécurités.

3.3. Redondance d'API gérant la commande et les sécurités

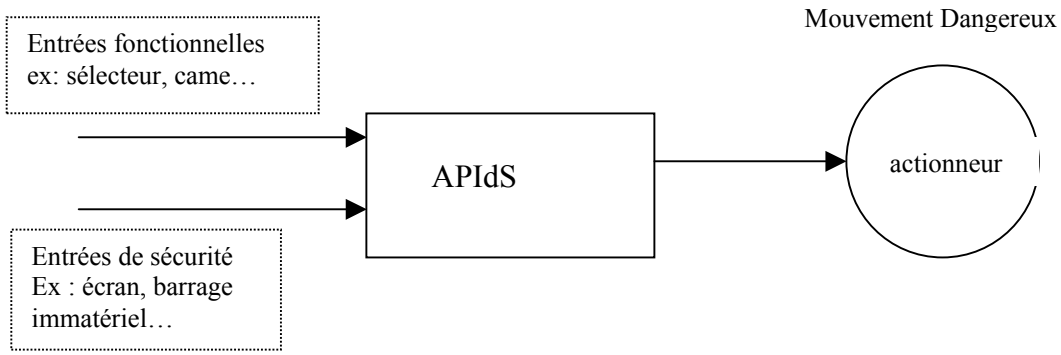
Dans cet exemple, la redondance choisie repose sur la mise en œuvre de deux automates devant donner la même information pour que celle-ci soit prise en considération ; en cas de discordance, il y a arrêt du processus et mise en sécurité.
La redondance permettant une meilleure disponibilité, c'est-à-dire celle où l'information d'un seul des API suffit pour commander le mouvement, ne permet pas d'assurer un bon niveau de sécurité en présence de la défaillance d'une des deux voies. Elle ne sera pas analysée dans ce document.



Dans ce cas, la défaillance de l'un des deux automates ne peut pas mener à l'accident. La discordance entre les deux sorties des API est détectée par un circuit extérieur qui commande l'arrêt du mouvement dangereux et interdit la remise en fonctionnement obligeant la réparation de l'API défaillant.
Cette architecture pourra être utilisée pour des systèmes où il est admis que la probabilité de défaillance de mode commun des deux API est négligeable. Elle doit être mise en œuvre par des spécialistes capables de valider l'application sachant que la sécurité dépend des mesures prises pour réduire les défaillances de mode commun et de la validation du circuit traitant la discordance des sorties des API.

3.4. APIdS gérant la commande et les sécurités

Cette architecture peut être rencontrée sur les systèmes où la commande et les sécurités sont fortement imbriquées, comme la commande de certaines machines telles que les presses mécaniques par exemple.



Sur cette représentation on remarque qu'une défaillance de l'APIdS pourrait conduire directement à l'accident malgré les sécurités initialement prévues. Il faut préciser toutefois qu'un APIdS a justement été construit pour qu'une défaillance matérielle ou une mauvaise conception du logiciel système conduisant à une commande intempestive soit peu probable par rapport à celle d'un API standard. Resteront toutefois à traiter comme pour les autres solutions, les problèmes liés aux programmes applicatifs, aux câblages, à la validation et à la maintenance.

3.5. APIdS gérant les sécurités séparées

Lorsque la complexité ou le nombre de fonctions de sécurité à traiter est important, il est envisageable de gérer les fonctions de sécurité par un APIdS séparé de la commande fonctionnelle de la machine. Dans cette architecture comme dans celle où l'APIdS devait gérer en plus le fonctionnel, la sécurité du système repose entièrement sur le comportement de l'APIdS en présence de défaillances. L'avantage de cette solution par rapport à la précédente, réside dans le fait que n'ayant à gérer que les sécurités, la validation s'en trouve simplifiée. De plus, les modifications du fonctionnel n'ont pas de conséquences sur le traitement des sécurités.

3.6. Conclusion

Le tableau ci-après résume les principales architectures décrites et potentiellement rencontrées sur un équipement industriel.

Paragraphe	Architecture	Commentaires
§ 3.1	Un API	Déconseillé pour la gestion des fonctions de sécurités.
§ 3.2	**API + traitement des sécurités séparées**	**Conseillé lorsque le traitement des sécurités ne nécessite pas d'opérations complexes. Seul le traitement des sécurités séparées est à valider.**
§ 3.3	Redondance d'API	Validation complexe nécessitant une expertise.
§ 3.4	Un APIdS gérant le fonctionnel et les sécurités	Validation nécessitant une expertise. (Point abordé dans les paragraphes suivants) limité à certaines applications bien ciblées.
§ 3.5	Sécurités séparées traitées par un APIdS	idem

L’analyse de ce tableau montre la diversité des architectures qui s'offre aux concepteurs de circuits de commande de machines pour gérer la sécurité. Lorsque cela est possible, il convient de retenir les architectures où "les sécurités sont séparées du fonctionnel". Cette solution a l’avantage d’identifier avec précision l’ensemble des moyens mis en œuvre pour éviter les situations à risque pouvant conduire à l’accident, mais aussi de bien circonscrire ce qui doit être testé et validé. Néanmoins, malgré ce choix de structure où le traitement des sécurités est séparé du circuit de commande, les difficultés tant de conception que de validation vont dépendre de la complexité des fonctions à traiter ainsi que de la technologie utilisée pour réaliser ces sécurités.

- *La logique câblée*

Cette technologie à base de relais électromécaniques, a fait ses preuves depuis plusieurs décennies car il est aisé avec celle-ci de réaliser et de valider les fonctions de sécurité en respectant les catégories de l'EN 954-1 demandées pour les différents types d’applications rencontrés en machinerie [5] (ED 807).

- *Les blocs logiques de sécurité*

On trouve souvent en machinerie les mêmes fonctions destinées à assurer la sécurité (arrêt d’urgence, double commande, etc.). Depuis quelques années, des fabricants proposent des blocs pré-câblés réalisant ces fonctions. Ces blocs peuvent être à base de composants électromécaniques ou électroniques. Ils ont été conçus pour la seule fonction qu’ils doivent réaliser et ont été validés ou certifiés par un organisme tiers reconnu compétent. L’utilisateur doit les câbler conformément aux instructions du fabricant et il ne lui restera qu'à valider ou à faire valider leur agencement dans l'application.

- *Les dispositifs électroniques programmables*

Pour les fonctions plus complexes où l’électromécanique ne convient plus, il est possible d’utiliser des logiques à base d’électronique programmable. Mais là se pose le problème de la conception et de la validation qui demandent aux concepteurs plus de compétences et des moyens d'investigation plus importants. Cette technologie utilisée pour la sécurité n'est pas encore stabilisée. Pour y remédier, des fabricants d'API proposent des APIdS (Automate Programmable Industriel dédié à la Sécurité) qui devraient simplifier la conception d’un système réalisé à base de ce type d’équipement.

Dans la suite du document, nous allons aborder l'utilisation des APIdS et surtout les problèmes liés à la validation des applications qu'ils gèrent. Comme nous l'avons vu précédemment, ils peuvent être utilisés pour traiter les seules fonctions de sécurité mais leur puissance de traitement peut permettre de traiter en même temps le fonctionnel.

4 - AUTOMATES PROGRAMMABLES DEDIES A LA SECURITE (APIdS)

Ces automates se distinguent des API standards par la mise en œuvre de moyens spécifiques qui leurs permettent de répondre de manière définie à l'apparition d'une défaillance d'un de leur composant.

Deux grandes classes cohabitent :

a) Les APIdS orientés vers la commande de processus tels que : Tricon de Triconex, H51 de Hima, 5000S de AEG Schneider Automation,…
b) Les APIdS orientés vers la commande des machines tels que : 95F, 115F et la série 400 F de Siemens, PSS 3000 et 3056 de Pilz, ABB Master 220/1,…

Les premiers sont conçus pour assurer la disponibilité d'un processus c'est-à-dire qu'ils ont pour mission de poursuivre le process en cours en toute sécurité malgré la défaillance d'une voie de traitement.

Les seconds sont orientés sécurité machine et ils doivent interrompre un mouvement dangereux dès qu'une voie de traitement est défaillante. Leurs temps de réponse sont beaucoup plus courts que les APIdS orientés vers la commande de processus.

Cette différence est fondamentale car elle a une influence évidente, tant sur l'architecture interne des APIdS concernés que sur le contenu des logiciels applicatifs.

4.1. APIdS en commande de processus

Ces automates mettent en œuvre des architectures redondantes d'ordre 3 avec voteur ou une architecture d'ordre 2 avec détection des fautes du canal défaillant par des autotests. Seules ces structures sont capables d'une part, de détecter la voie défaillante pour initialiser une procédure d'urgence ou d'alerte permettant la remise en état et d'autre part, de poursuivre le processus en maintenant l'efficacité des sécurités.

Le schéma N° 1 [6] (EXERA) décrit ci dessous un exemple d'architecture interne d'un APIdS utilisé en sûreté des processus à savoir trois voies indépendantes et un voteur.

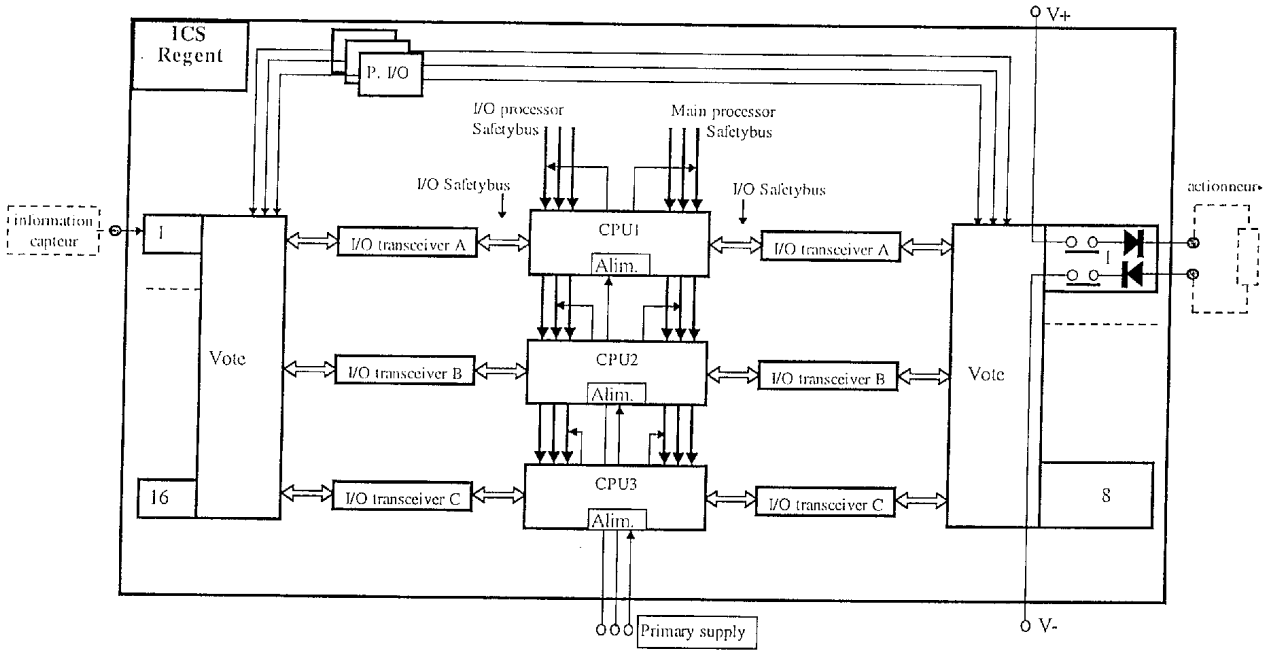


Schéma n° 1

4.2. APIdS en commande de machine

Ces automates peuvent se contenter d'architectures redondantes d'ordre 2 avec comparateur permettant de vérifier que les deux voies, à partir des mêmes informations d'entrée, donnent les mêmes résultats en sortie.

En réalité, les constructeurs d'APIdS dédiés à la machinerie ont développé pour certains des structures redondantes d'ordre 2 et pour d'autres des structures tri-redondantes. De même, il existe des structures à voies indépendantes ou communicantes ou encore des structures utilisant les mêmes composants ou au contraire des composants différents nécessitant des logiciels applicatifs différents ou non.

Ces différentes solutions montrent la diversité des moyens utilisés et surtout que chaque solution n'est qu'un compromis privilégiant tel ou tel paramètre de la sécurité comme par exemple :

- la rapidité de réaction face à une défaillance,

- la réduction de l'influence des pannes de mode commun,
- la détection des pannes latentes,
- le temps réponse de l'application.

Le schéma n° 2 [6] (EXERA) présente une structure redondante d'ordre 2 de l'unité centrale de l'automate. Il est à remarquer que les deux voies sont indépendantes et qu'aucun contrôle n'est effectué entre les CPU A et B. L'intérêt d'une telle architecture est d'éliminer la contamination d'une voie par l'autre car elles sont complètement indépendantes. L'inconvénient tient au fait que seules les défaillances ayant une influence sur la sortie sont détectées par le comparateur, les autres défaillances (pannes latentes) ne sont pas traitées.

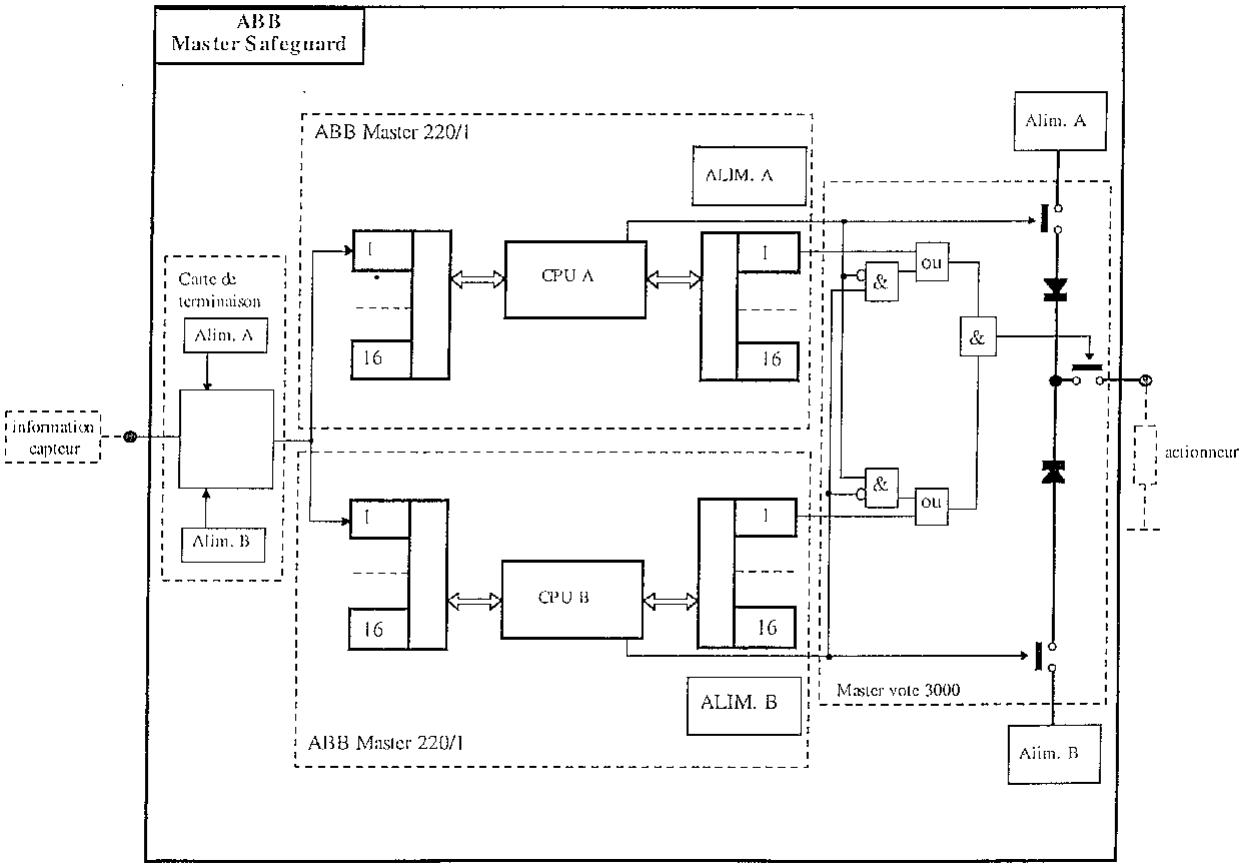


Schéma n° 2

Le schéma n° 3 [6] (EXERA) présente une structure tri-redondante d'un APIdS utilisable en machinerie. Dans ce cas, le constructeur a choisi de procéder à des échanges inter voies ce qui permet la détection de certaines défaillances latentes qui ne seraient pas détectées par le comparateur de sortie.

Cette solution a toutefois l'inconvénient d'être une source potentielle de pannes de mode commun car les voies ne sont plus totalement indépendantes [4].

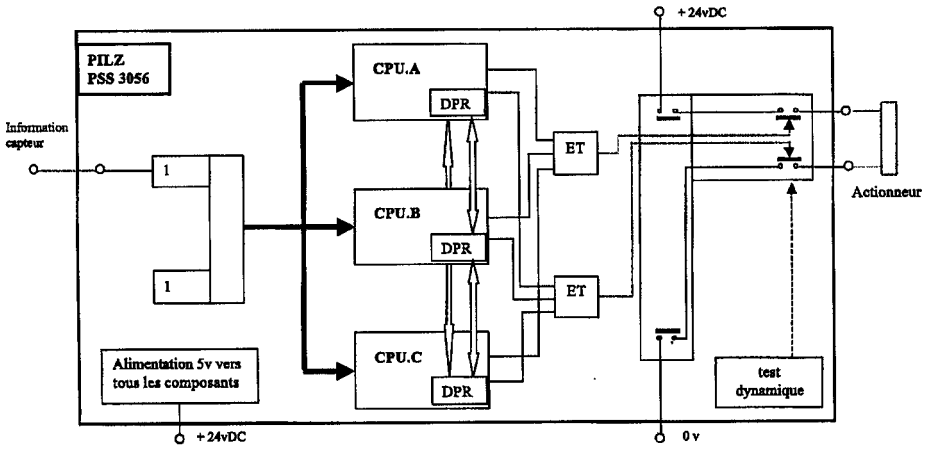


Schéma n° 3

5 - GESTION DES FONCTIONS DE SECURITE PAR APIDS CONÇUS POUR LA MACHINERIE

Les problèmes de l'utilisation d'un APIdS pour gérer les fonctions de sécurité doivent être abordés suivant trois aspects :

a) l'APIdS en tant que matériel,
b) le programme applicatif dont il doit être muni,
c) l'interface "APIdS – machine" c'est-à-dire l'interconnexion de l'automate avec l'équipement qui lui aussi revêt une part importante dans la réussite des objectifs fixés.

5.1. Aspect matériel

Ces automates mettent en œuvre des moyens matériels qui leur permettent de répondre de manière définie (pannes orientées) à l'apparition d'une défaillance d'un de leurs composants et l'on peut citer :

- une structure au moins redondante des principaux éléments matériels ou autres dispositions donnant une garantie au moins équivalente (dynamisme, contrôle,…),
- une exécution contrôlée des logiciels systèmes et applicatifs dans des temps limités,
- des logiciels applicatifs pré-écrits, des blocs de fonctions pré-certifiés et ou pré-validés,

- une série d'autotests destinés à vérifier l'absence de défauts latents (par exemple au niveau, des mémoires EPROM en lecture, des RAM en écriture et en lecture, des microprocesseurs par la vérification de l'exécution d'instructions de contrôle, des horloges, des alimentations…),
- une certification ou une validation du produit par un organisme compétent.

Rappelons que des organismes compétents européens (BIA, BG, TÜV...) valident suivant une démarche volontaire des APIdS en fonction de différents référentiels issus :
- d'une norme européenne harmonisée: la EN 954-1 [7],
- de normes internationales de la CEI (Commission Electrotechnique Internationale) : la CEI 61508 et la CEI 62061 (encore à l'état de projet) [8] [9],
- d'une norme nationale, par exemple la VDE 801 pour l'Allemagne.

Il faut toutefois remarquer qu'il n'existe pas de norme spécifique APIdS contrairement aux normes API qui permettent d'avoir un avis de conformité sur ce produit [10].
A ce jour, un APIdS composant matériel sans son logiciel applicatif, n'est pas considéré au sens réglementaire comme un composant de sécurité pouvant être mis isolément sur le marché. Les certificats délivrés pour certains APIdS n'étant pas des attestations d'examen CE de type (non listés à l'annexe IV de la directive Machines), on peut tout au plus en déduire une présomption d'aptitude à gérer des fonctions de sécurité. Cette présomption sera d'autant plus forte si l'organisme est reconnu pour la qualité de ses expertises dans ce domaine.

5.2. Aspect logiciel

Un APIdS sans son logiciel applicatif n'a aucune fonction définie. C'est uniquement lorsqu'il exécute un logiciel applicatif spécifique qu'il devient apte à gérer une ou plusieurs fonctions de sécurité d'une application industrielle. Cette propriété justifie l'intérêt des APIdS, car il devient ainsi possible avec un seul type de composant matériel et divers logiciels applicatifs de réaliser l'ensemble des fonctions de sécurité nécessité par la diversité des applications en automatisme. De plus, la possibilité de modifier le logiciel permet une évolution de l'application comme par exemple la gestion des zones de protection évolutives dans le temps.
Rappelons que le logiciel applicatif est le logiciel développé avec le langage propre à chaque APIdS pour gérer une application.
Cas particulier : Pour les applications de type presse ou machine à bois, pour lesquelles le fonctionnel de la machine a un rôle déterminant sur la sécurité, le logiciel applicatif inclura fonctionnel et gestion des sécurités.

Quant au logiciel système (qui gère le fonctionnement interne de l'APIdS), il n'est pas accessible aux utilisateurs. Ayant été validé en même temps que la partie matérielle de l'APIdS, il n'intervient pas sur la validation du logiciel applicatif.
Brièvement on peut citer les étapes nécessaires pour valider un logiciel applicatif :

a) S'approprier les moyens mis en œuvre par le développeur pour atteindre l'objectif de sécurité revendiqué en s'appuyant sur :
   - l'existence de prescriptions fonctionnelles de la machine (exigences normatives, de sécurité, de contrôles...),
   - la façon dont ces prescriptions ont été mises en œuvre,
   - les contrôles et évaluations réalisés (auto certification ou certification par un organisme compétent),
   - l'existence d'une notice d'utilisation spécifique à l'application.

b) Vérifier de façon purement formelle que le logiciel est bien écrit :
   - modularité,
   - hiérarchisation des modules,
   - nombre d'instructions par module,
   - nombre d'entrées/sorties des modules,
   - affectation des entrées et des sorties,
   - commentaires,
   - ...

En fait, à partir d'outils spécifiques on doit savoir si le logiciel a été correctement écrit pour qu'il soit lisible, maintenable et testable. Ce critère est nécessaire mais n'est néanmoins pas suffisant pour valider un logiciel, car on ne sait pas encore à ce stade ce qu'il exécute réellement.

c) Vérifier que le logiciel est conforme aux spécifications définies dans le cahier des charges.
   Pour y satisfaire, il est nécessaire de stimuler l'APIdS afin de vérifier que sa réaction est conforme à celle spécifiée, et cela dans toutes les configurations possibles d'utilisation.
   En théorie, il faut vérifier la réponse de l'APIdS avec son logiciel applicatif pour chaque séquence d'entrée.
   En réalité, on se rend compte rapidement qu'un test exhaustif devient irréalisable si le nombre de fonctions ou de séquences est important. Il convient alors d'utiliser des

méthodes spécifiques adaptées aux logiciels pour assurer un niveau de confiance raisonnable quant à la conformité du cahier des charges.

d) Vérifier la pérennité de la solution retenue

Le contrôle étant réalisé, il faut s'assurer que des modifications de programme ne pourront pas être réalisées sans exécuter une procédure spécifique destinée à maîtriser cette modification. Celle ci doit être validée et surtout inscrite dans un processus de traçabilité ce qui peut par ailleurs limiter la flexibilité et la souplesse reconnue à un APIdS.

En ce qui concerne le développement du logiciel applicatif, les résultats d'une étude en cours à l'INRS donneront de plus amples informations.

5.3. Aspect intégration dans l'équipement

Cet aspect ne sera que brièvement abordé car il ne diffère que très peu des applications à base de logique câblée dont on maîtrise assez bien la mise en œuvre et la validation. Il faut toutefois signaler que cette mise en œuvre n'est pas commune à tous les APIdS et que chaque fabricant propose sa manière de bien réaliser cette interconnexion suivant la catégorie (EN 954-1) revendiquée pour l'application (voir référence câblage des APIdS) [11].

Partant d'un APIdS avec son logiciel applicatif validé, le constructeur ou l'intégrateur doit le connecter à sa machine de façon sûre.

Pour cela, il doit :

a) Choisir des capteurs et des actionneurs compatibles avec le niveau de sécurité attendu et le logiciel applicatif mis en œuvre dans l'APIdS. Ils seront soit auto-contrôlés, soit à sécurité intrinsèque, soit doublés selon le type de capteurs/actionneurs retenus et le niveau de sécurité revendiqué.

b) Réaliser le câblage entre les différents capteurs/actionneurs et les entrées/sorties de l'APIdS comme conseillé par le fabricant de l'APIdS et suivant le type de carte d'entrées/sorties utilisé.

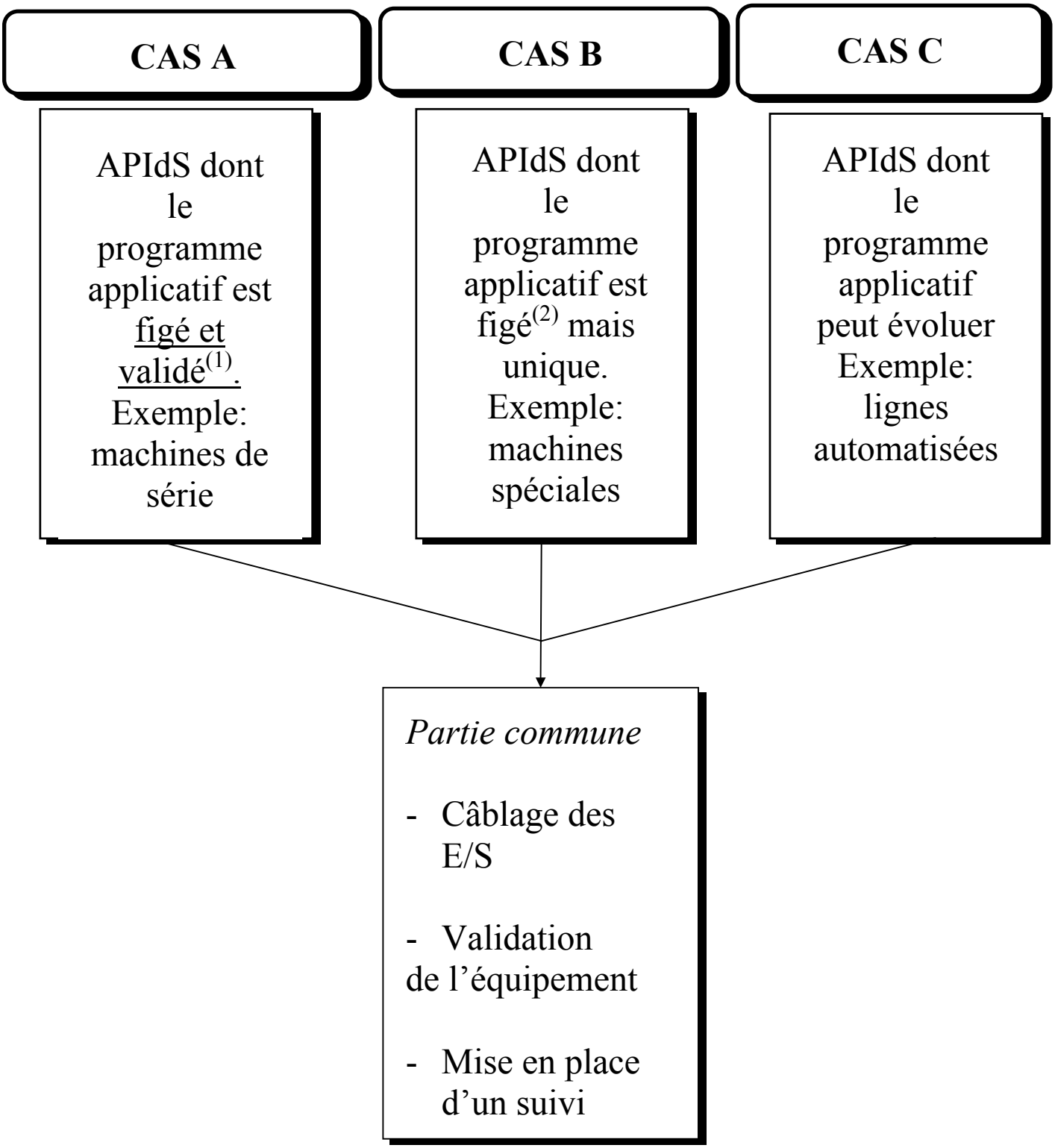
c) Valider la réalisation globale :
   - vérifier que toutes les fonctions prévues répondent au cahier des charges,
   - injecter s'il y a lieu des fautes sur les capteurs et actionneurs de la machine ainsi que sur le câblage de ceux-ci et s'assurer à chaque fois du bon comportement de la machine.

d) Etablir la notice d'utilisation et de dépannage de la machine ainsi que les procédures de contrôle à mettre en place tout au long de son cycle d'utilisation pour en assurer sa pérennité.

6 - CLASSEMENT DES APPLICATIONS GEREES PAR APIDS

Le synoptique suivant propose une classification des diverses applications gérées par APIdS selon trois familles.

Classement des applications gérées par APIdS en machinerie



(1) : Validation par le constructeur de l'APIdS ou par un organisme compétent.

(2) : Le programme applicatif est crée par le constructeur de la machine. Sa validation intervient en fin de cycle de développement de la machine.

- *La première famille* (cas A) concerne les machines autonomes mettant en œuvre peu d'entrées/sorties et dont les fonctions logiques à réaliser sont assez simples bien qu'étant séquentielles (presses, presses plieuses, cisailles, machines à bois classiques…). Dans ce cas particulier où l'équipement possède une fonction bien définie, il devient possible de figer son logiciel applicatif, de le protéger contre toutes modifications non contrôlées et ensuite de le dupliquer sur tous les équipements pour lesquels il a été développé. L'avantage d'une telle procédure réside dans le fait qu'un seul logiciel est à développer, à mettre au point et à valider.

Dans cette première famille, le programme applicatif est défini et figé sous la forme d'un module validé pour lequel tous les paramètres internes sont fixés ainsi que l'affectation des entrées et des sorties. Ainsi, l'intégrateur n'a plus qu'à câbler l'APIdS à sa machine en respectant le plan de câblage fourni avec le logiciel applicatif. Il lui restera toutefois à contrôler par un test fonctionnel la bonne réalisation du câblage. Le concepteur du logiciel applicatif devra lui fournir les tests à effectuer pour l'aider à réaliser cette vérification.

En fait cela est similaire à la philosophie des blocs logiques de sécurité pour lesquels l'intégrateur a pour seule initiative la réalisation du branchement et le contrôle de la bonne mise en œuvre sans se préoccuper des problèmes liés a la réalisation technique.

Bien entendu le logiciel applicatif devra être verrouillé de façon à ce qu'il ne puisse plus être modifié par l'utilisateur et il devra comporter une signature garantissant sa pérennité tout au long de son utilisation.

Quelques constructeurs (Pilz, Siemens) proposent déjà des logiciels applicatifs validés.

Pour des machines à risques élevés, il nous semble judicieux de confier la validation de l'ensemble de l'application à un organisme reconnu pour ses compétences en la matière.

Pour des machines à faibles risques, le constructeur pourra auto-certifier son produit directement à condition de respecter les étapes énoncées ci-dessus. S'il n'en a pas les capacités, il devra utiliser des technologies éprouvées et connues ou faire appel à un organisme reconnu.

On peut remarquer dans ce type d'application où le programme est verrouillé, que l'utilisateur final n'a en aucun cas la possibilité d'intervenir sur le programme applicatif donc sur la gestion des sécurités. Seuls les paramètres de la machine (sans incidence sur la sécurité) lui sont accessibles. Pour assurer la pérennité des fonctions de sécurité, toute modification du processus de travail ou tout dépannage nécessitant une modification du programme devra faire l'objet d'une demande d'intervention auprès de l'intégrateur, charge à ce dernier de faire le nécessaire et de revalider l'équipement.

- *La deuxième famille* (cas B) rassemble l'ensemble des machines spéciales développées soit unitairement, soit en série limitée. Contrairement au cas A, ces applications possèdent des logiciels applicatifs non standard. L'utilisateur ou l'intégrateur développe son propre logiciel applicatif. Ensuite il devra le verrouiller pour éviter toute modification et le valider ou le faire valider sachant que cette validation ne correspondra qu'à cette application.

Les problèmes soulevés dans ce type d'application sont :

- la nécessité d'un personnel hautement qualifié en programmation et sécurité,
- la difficulté pour l'exploitant à maîtriser la validation,
- le coût d'une telle validation du fait de son unicité et des moyens à mettre en œuvre.

Compte tenu de ces problèmes, ces applications gérant des fonctions de sécurité seront généralement réservées à des grandes entreprises sur des installations complexes.

- *La troisième famille* (cas C) se distingue des deux précédentes par le fait que le logiciel applicatif de l'APIdS gérant les fonctions de sécurité doit pouvoir être facilement adapté aux évolutions d'une production automatisée rencontrées par exemple dans l'industrie automobile, alimentaire ou la fabrication de produits en béton dans le bâtiment et les travaux publics. Cette obligation contraint l'intégrateur à fournir un système ouvert ne lui permettant pas de garantir une sécurité pérenne, contrairement aux cas A et B où le logiciel applicatif gérant les fonctions de sécurité est validé et verrouillé pour l'application.

Cette grande souplesse de modification du programme utilisateur pose des difficultés quant à la gestion et au maintien de la sécurité après une modification. En effet, de la même façon que chaque application nécessite une conception et une validation qui lui est propre, chaque modification apportée doit aussi être répertoriée et validée. Ceci demande un personnel hautement qualifié en programmation et l'existence de procédures de modifications à mettre en œuvre et à respecter.

7 - CONCLUSION

Après avoir montré la diversité des architectures pouvant être mises en œuvre, fait l'inventaire des différents types d'APIdS présents sur le marché, répertorié les types d'applications, il est à noter que l'utilisation du composant APIdS pour résoudre les problèmes de sécurité d'une application n'est pas une condition suffisante et qu'il faudra comme pour toute application valider l'ensemble du système.
Ce qui pose problème aujourd'hui dans l'usage d'un APIdS n'est pas le composant en tant que tel mais plutôt la complexité et la validation de la mise en œuvre tant du point de vue logiciel applicatif que du câblage des capteurs et surtout des actionneurs qui lui sont associés.

Aujourd'hui, on peut admettre que les machines équipées d'un APIdS avec son programme applicatif figé (cas A et B), verrouillé et validé par un organisme compétent apporte une garantie suffisante pour un fonctionnement en sécurité.
Pour les réalisations d'équipement à base d'APIdS dont le programme applicatif est ouvert (cas C) permettant ainsi les évolutions ultérieures, c'est à l'intégrateur ou à l'utilisateur à apporter la preuve du niveau de sécurité revendiqué, de surcroît il doit aussi en assurer la pérennité tout au long du cycle de vie du système. Pour ce cas, devant la difficulté du problème à résoudre, il est recommandé d'avoir encore recours dans la mesure du possible aux solutions classiques ayant fait leurs preuves (logique câblée, bloc logique, redondance…). Si la solution par APIdS est incontournable, il conviendra pour garantir une bonne mise en œuvre que les intégrateurs ou utilisateurs qui n'ont pas le personnel qualifié, se fassent assister par un organisme compétent. Celui-ci les aidera à valider le produit final, mais aussi les conseillera depuis la conception, la mise en œuvre, l'exploitation et jusqu'à la fin de vie de l'installation.

REFERENCES BIBLIOGRAPHIQUES

[1] Les automates programmables - Cahiers de Notes Documentaires n°117, 4ème trimestre-1984, pp. 467-474.

[2] Note établie par le bureau CT5 du Ministère de l'Emploi et de la Solidarité, note relative à l'acceptation de certains automates programmables pour gérer des fonctions de sécurité sur machines (26 Mai 1998).

[3] NF EN 292 Sécurité des machines - Notion fondamentales, principes généraux de conception : Partie 1 : Terminologie de base, méthodologie, AFNOR, Paris, 1991, 33 p.

[4] Outil pour l'évitement des fautes logicielles - défaillances de mode commun dans les systèmes de sécurité, projet Européen STARCES SMT4CT97-2191, mars 2000, annexes au rapport final, 54 p.

[5] Sécurité des machines et des équipements de travail - Moyens de protection contre les risques mécaniques. Paris, INRS, ED 807, 2e édition (2000).

[6] Document EXERA, Groupe de travail "Systèmes de sécurité à API-APIdS : panorama", rapport EXERA (rédacteur : J.F. AUBRY), Paris, décembre 2000, 68 p.

[7] NF EN 954-1 (2/1997) Sécurité des machines- Parties des systèmes de commande relatives à la sécurité. Partie 1 : Principes généraux de conception. Paris- La défense, AFNOR.

[8] CEI 61508 "Sécurité fonctionnelle des systèmes électriques / électroniques / électroniques programmables relatifs à la sûreté".

Partie 1 : Prescriptions générales, UTE C 46-061, avril 2000, 59 p.
Partie 2 : Exigences pour les systèmes électriques / électroniques / électroniques programmables, UTE C 46-062, avril 2000, 71 p.
Partie 3 : Prescriptions concernant les logiciels, UTE C 46-063, avril 2000, 49 p.
Partie 4 : Définitions et abréviations, UTE C 46-064, avril 2000, 26 p.
Partie 5 : Exemples de méthodes pour la détermination des niveaux d'intégrité de sécurité, UTE C 46-065, avril 2000, 28 p.
Partie 6 : Guide pour l'application des parties 2 et 3, UTE C 46-066, avril 2000, 75 p.
Partie 7 : Bibliographie des techniques et des mesures, UTE C 46-067, avril 2000, 115 p.

[9] CEI 62061 "Sécurité des machines – Sécurité fonctionnelle des systèmes Electriques / Electroniques / Electroniques Programmables" – Version CD, septembre 2000, 63p.

[10] EN/CEI 61131 - Automates programmables, Octobre 1992
Partie 1 : Informations générales.
Partie 2 : Spécifications et essais des équipements.
Partie 3 : langages de programmation.
Partie 4 : Recommandation à l'utilisateur.
Partie 5 : Communications.
Partie 6 : (à l'étude).
Partie 7 : Programmation en logique floue.

[11] Document CRAMIF - Utilisation d'Automates pour l'exécution de fonctions de sécurité. Complément d'information à destination des concepteurs/intégrateurs d'installations automatisées et des rénovateurs d'équipements de travail. (En cours d'édition).