Vérifiez-vous vos programmes API ? Non.

Tout automaticien n'aime pas trop que l'on mette le doigt où ça fait mal. Et s'il existe encore des points noirs en terme de programmation, c'est bien celui de la vérification des programmes. Notamment en ce qui concerne le suivi de règles claires.

Depuis la création des automates programmables, il reste un sujet qui n'a guère évolué, c'est celui de la programmation. Inutile de refaire l'historique des outils de programmation, mais ils ont été développés pour la plupart, par des fournisseurs d'automates, et donc ils sont propriétaires. Et si en une vingtaine d'années la communication s'est ouverte permettant à des automates de diverses marques de dialoguer entre eux, pour la programmation rien de tel. Et le jour n'est pas encore venu ou une programmation prévue pour un automate Siemens pourra être introduite dans un automate Rockwell.

Ce constat, tous les utilisateurs l'ont fait. Certaines normes sont venues clarifier le marché, ouvrir les portes à d'autres outils, mais il n'empêche que la programmation et l'utilisation de ces outils de programmation reste l'oeuvre d'un automaticien qui doit respecter des règles. Pour les vérifier, le seul outil qui reste, c'est le regard d'un autre automaticien.

Et comme dans ce monde globalisé, il devient difficile d'avoir des automaticiens ayant le recul et l'expérience nécessaire, le sujet de la programmation des automates reste un sujet épineux.

LE CAS EXEMPLAIRE DE PSA

Or, les utilisateurs voient la pression augmenter, et comme le précise Laurent Mauguy de PSA qui reste l'un des précurseurs en la matière « *les automatismes ne doivent pas être sur le chemin critique d'un projet* ». En clair, lorsque l'on demande aux équipes de production de développer un nouvel outil pour produire un véhicule nouveau, il est hors de question que le frein soit celui de la programmation des automates, et il ne s'agit pas en l'espèce de la livraison du programme, mais bien de la fiabilité du programme lors de la montée en puissance de la ligne, devoir débogger le programme ou le maintenir ensuite, ne doit pas entraîner un arrêt de la production. Surtout « *que nous avons une politique au sein du groupe qui nous fait travailler avec plusieurs intégrateurs, qui peuvent être étrangers et éloignés, et donc il nous faut suivre et valider leurs travaux à distance* ».

Un sujet qui se rajoute à une dégradation de la qualité des programmes, avec un turnover important chez certains ingénieristes. Aussi, pour limiter les risques lors de la réception d'un programme, le groupe français a mis au point la démarche « Actif » qui est dorénavant obligatoire pour tout nouveau projet que ce soit avec des automates Siemens, Schneider ou Rockwell.

Tout intégrateur ou fournisseur se devra de suivre les normes PSA comme la E03.65.038.G pour l'analyse fonctionnelle ou la E03.64.160.G pour le terminal opérateur. En final, lors de la programmation, le découpage des séquence devra se faire selon les règles PSA avec un nommage des variables spécifiques. Ces nommages seront les mêmes que ceux que l'utilisateur ou l'agent de maintenance trouvera sur le pupitre de travail, le programme, l'analyse fonctionnelle ou le terminal opérateur.

Ce programme Actif devenu la règle depuis deux ans, n'intervient que pour les nouvelles lignes, sur les anciennes, les règles précédentes perdurent.

Par exemple, dans les règles de conception et de réalisation, il est spécifié que pour chaque groupe de sécurité défini dans l'analyse fonctionnelle, un bit « groupe de sécurité OK » valide les traitements dépendant de ce « groupe de sécurité », ou pour la partie maintenance il est indiqué que « *tout programme est structuré et commenté* », chaque équation devra porter un commentaire d'entête indiquant la nature de l'équation, les commentaires devront être le plus précis possible et exprimés en termes liés à l'application et rédigés dans la langue du pays destinataire du programme. Le commentaire devra être d'un minimum de 10 caractères et de 150 au maximum »... sans parler de toutes les règles précises pour la génération des messages de défauts.

Des règles qui pour certains paraissent tellement évidentes qu'il serait intéressant de savoir comment font les autres industriels ? C'est bien là le problème. Evidemment, tout industriel a des règles plus ou moins formalisées, lorsqu'elles existent, aucun donneur d'ordres n'utilise le même système de règles.

UN OUTIL DE VÉRIFICATION...

Itris Automation Square qui a développé un outil sur lequel nous reviendrons, a dénombré une vingtaine de règles communes à pratiquement toute la profession et qui pourrait constituer le début du commencement d'un standard. Ce qui n'est rien vis-à-vis des 160 règles édictées par PSA et que

tout fournisseur se doit de suivre, sachant qu'à ces règles génériques se rajoutent des règles spécifiques à chaque famille de matériels.

Quelques grands intégrateurs ou ingénieristes ont édicté leurs propres règles qu'ils tentent d'imposer à leurs clients. Mais, encore une fois, chacun d'entre eux travaille dans son coin.

Face à cette complexité de règles à suivre, le groupe automobile se trouve en partie démuni lors de la réception des programmes. Comment valider qu'un programme a bien suivi toutes les règles ? Impossible de tout rejeter en bloc, mais valider le programme correspond à dédouaner le développeur qui part avec son chèque, et en cas de problème lors de la montée en puissance de l'application, c'est souvent un peu tard.

D'où l'idée de trouver un outil logiciel de validation des programmes. Seulement, l'offre n'est pas pléthorique, bien au contraire, on se rapproche du vide sidéral. « *Les fournisseurs d'automates programmables auraient pu développer et proposer de tels outils, mais aucun ne l'a fait* » se résigne Laurent Mauguy.

Sur ce marché, une PME, dans les années 90, avait développé à usage interne un outil permettant de valider quelques règles simples. C'est vers eux que PSA s'est tourné, donnant en quelques sortes à Itris Automation Square les moyens de se focaliser sur un marché que Eric Pierrel, Directeur Marketing d'Itris Automation Square, estime entre 25 et 50 millions sur le plan mondial.

Un vrai business pour cette PME de 14 employés qui grâce à son langage Glips a décidé d'aller sur ce créneau. Après import, le programme automate passe par une case d'abstraction de haut niveau basée sur le langage Glips, et il devient possible de contrôler le programme automate avec la vérification des règles de codage ; en quelque sorte un correcteur grammatical d'un traitement de texte, acceptant des textes issus d'open source ou de Microsoft Office.

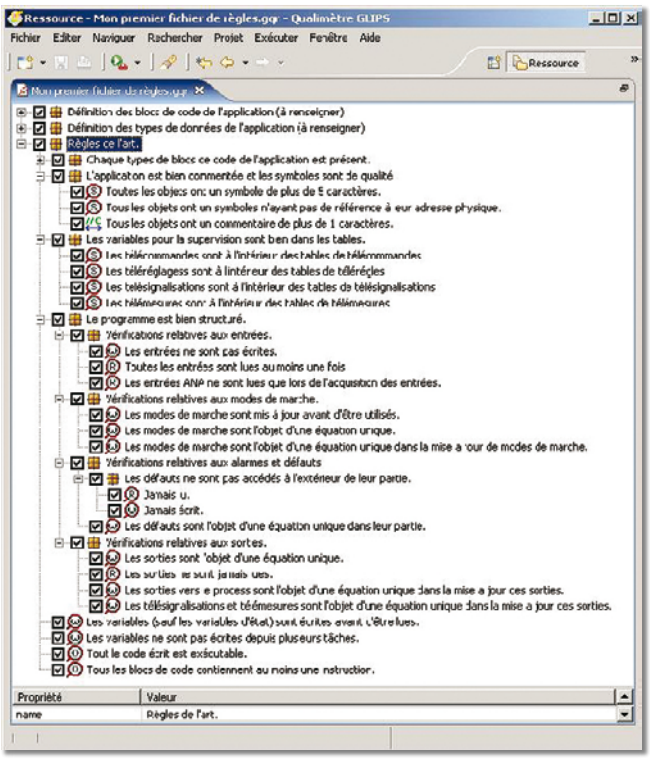
Ce logiciel du nom de PLC Checker, s'il vérifie la rigueur du programme n'a pas vocation à vérifier la conformité fonctionnelle du programme, ni à valider les contraintes temporelles, même si un programme bien écrit devrait aider à améliorer ces derniers critères.

Le quidam qui souhaite utiliser PLC Checker, devra se contenter d'une vingtaine de règles comme nous l'avons évoqué plus haut, ensuite il demandera à l'éditeur Itris de formaliser de nouvelles règles.

C'est ce qu'a fait PSA en demandant d'intégrer certaines des règles propres dans le logiciel. Aujourd'hui, ce sont 70 règles PSA qui sont prises en compte sur les 160, une dizaine de nouvelles devraient compléter l'offre d'ici un an. Certaines règles restent complexes à formaliser dans l'outil. Mais il est déjà impossible de faire passer pour « bon » un programme mal nommé, des variables inexistantes, des codes morts ou des sous-routines non appelées, comme c'est trop souvent le cas.

...QUI MONTRE QUE 100 % DES PROGRAMMES COMPORTENT DES ERREURS

La commercialisation du logiciel d'Itris utilise un schéma peu ordinaire. En effet, aucune licence d'exploitation, mais uniquement une interface pour le client lui permettant de télécharger son programme automate et quelques minutes après, un compte-rendu avec les erreurs fatales et les warning. Un paiement à l'acte qui pousse à bien travailler dès le début, et a permis, au sein d'une même équipe de développeurs, de voir des différences sensibles entre automaticiens.



PSA de son côté a demandé que deux types d'offres soient proposées à ses intégrateurs, l'une baptisée Qualité et qui autorise autant de contrôles que nécessaire pour être conformes, et l'autre Audit qui ne permet que deux contrôles pour un même automate, une offre qui va être abandonnée, car trop limitée. Si au démarrage, il semblait opportun de proposer une offre alléchante en terme de prix, il a bien fallu se rendre à l'évidence, le nombre de règles non-suivies était tel qu'il était impossible en deux aller-retour de parvenir au résultat demandé.

PSA s'est même « amusé » à éditer un premier tableau sur le nombre d'erreurs fatales dans un programme. Le terme « fatal » impliquant un refus de valider le programme, car non-conforme, à l'ensemble des règles édictées. 56 % des programmes avaient plus de 1070 erreurs, vous avez bien lu, et seulement 29 % des programmes avaient moins de 535 erreurs.

En moyenne, ce sont 12 contrôles qui sont nécessaires pour qu'un programme passe les tests de l'offre Qualité. Pour les programmes bons du premier coup, il faudra attendre encore quelques années, mais PSA veut pousser ses fournisseurs à travailler correctement dès le début, c'est ainsi qu'un tableau récapitulatif des bons et des mauvais de la classe sera publié l'année prochaine, ça promet.

Restera pour Itris à promouvoir son produit auprès d'autres entreprises françaises ou étrangères. Sans nul doute, l'une des prochaines usines de PSA construite en commun avec Mitsubishi en Russie devrait donner à la PME française une ouverture vers l'extérieur ■