Solutions

INFORMATIQUE EMBARQUÉE

# De nouveaux outils facilitent la certification logicielle

**Pour faire face à l'explosion de la complexité des programmes, les industriels des secteurs aéronautique et ferroviaire ont mis en place des procédures de certification logicielle. Si ces dernières apportent la garantie d'un certain niveau de sûreté de fonctionnement, elles sont néanmoins longues et coûteuses. Pour pallier ces inconvénients, les éditeurs d'outils et de composants logiciels proposent désormais des kits de certification pour leurs produits. Les industriels d'autres secteurs, notamment ceux de l'automobile ou du médical, pourraient également tirer d'importants bénéfices de ces outils.**

En 1978, le Mirage 2000 fut le premier avion de série à confier certaines fonctions à un calculateur plutôt qu'à des actionneurs mécaniques. Puis l'année 1984 vit le lancement de l'*Airbus* A320, premier avion commercial à disposer de commandes entièrement gérées par des calculateurs. Depuis lors, les calculateurs ne cessent de se multiplier, sans qu'à ce jour on ne déplore aucun crash d'avion lié à une défaillance logicielle. Le secteur automobile, lui aussi, effectue sa migration du "tout-mécanique" vers le "tout-électronique". Toutefois, dans l'automobile, le logiciel inquiète. Les constructeurs redoublent d'efforts mais des pannes inexplicables (les "bugs") continuent de survenir. Pour quelle raison peut-on avoir davantage confiance en un logiciel embarqué à bord d'un avion qu'à bord d'une voiture ?

Essentiellement parce que tous les logiciels avioniques sont soumis à des contraintes de certification. Selon Olivier Charrier, ingénieur principal pour les activités "aérospatial et défense" chez *Wind River*, « *certifier un équipement, c'est lui attribuer un certain degré de confiance. Concernant la partie logicielle d'un produit certifié, l'industriel doit prouver à un organisme de certification qu'il a respecté un processus de développement conforme à l'état de l'art dans le secteur concerné. C'est la raison pour laquelle on trouve différents standards selon les secteurs (aéronautique, automobile, ferroviaire, etc.).* »

Le standard de base, applicable à l'industrie au sens large, est l'IEC 61508. Il se décline en différentes normes : EN 50128 pour le ferroviaire, IEC 60880 pour le nucléaire, ECSS pour le spatial, ISO 26262 pour l'automobile, IEC 61511 pour les industries de process ou encore IEC 60601 pour le médical. Dans l'aéronautique, le standard a pour nom DO-178B aux États-Unis et ED-12B en Europe (même si, dans le langage courant, la plupart des industriels du secteur emploient la dénomination américaine).

À chaque secteur sa norme. Mais elles diffèrent selon leur caractère obligatoire ou non. En effet, seuls les industriels de l'aéronautique et du ferroviaire sont tenus de présenter leurs produits devant des organismes de certification (pour l'aéronautique il s'agit de l'EASA en Europe et de la FAA outre-Atlantique). Dans les autres secteurs, l'application des standards est seulement recommandée. Nous verrons d'ailleurs que certains industriels (principalement ceux de l'automobile et du médical) pourraient beaucoup gagner à s'inspirer des travaux réalisés dans l'aéronautique. Mais avant cela, il faut bien comprendre les problématiques liées à l'aéronautique. En effet, étant donné que la norme DO-178B est connue et utilisée depuis longtemps dans ce secteur, les éditeurs de logiciels, d'outils de développement et d'outils de tests ont lancé ces dernières années de nombreux produits facilitant les procédures de certification.

L'objectif de la certification aéronautique est de prendre en compte le remplacement des commandes mécaniques par des calculateurs pour la gestion de fonctions critiques, liées à la sécurité des passagers ou à la transmission des commandes de vol. En 1985, une communauté d'industriels a publié un document de référence, la DO-178A, qui impose notamment d'appliquer des tests unitaires sur tous les composants des programmes. Le traitement des tâches en temps réel n'étant pas suffisamment au point à cette époque, cette première norme imposait des développements sur le principe du séquenceur (avec des tâches effectuées dans un ordre prédéterminé). « *Il faut bien voir ces normes de certification logicielle comme un état de l'art : elles mettent à profit les méthodes de génie logiciel et les outils de développement disponibles sur le marché et maîtrisés à un instant t*, commente Thierry Billoir, responsable de l'activité aérospatial et défense chez *Geensys*. *Un exemple : le principe des applications "préemptives", qui prennent la main sur le système lorsque survient un certain type d'évènement. Cela n'a été introduit qu'avec la seconde version de la norme, la DO-178B, parue en 1992. De même, les tests unitaires, dont le rôle était primordial, ne sont plus considérés dans cette version B que comme un complément aux tests fonctionnels.* »


MathWorks

Les éditeurs proposent désormais des kits de qualification. Ceux-ci comprennent toute la documentation nécessaire pour constituer le dossier de certification.

## L'essentiel

- L'industrie aéronautique a développé ses propres standards et mis au point des procédures de certification. Mais la certification coûte très cher, et les procédures durent plusieurs années.
- Aujourd'hui, les éditeurs proposent des outils et des composants logiciels déjà qualifiés par les organismes de certification, ce qui facilite et raccourcit les procédures de certification.
- Dans les secteurs de l'automobile et du médical, les logiciels ne sont pas encore soumis à certification mais pourraient le devenir.

AdaCore

L'industrie aéronautique a toujours eu un rôle de précurseur dans le développement d'applications critiques. Elle n'hésite d'ailleurs pas à utiliser des méthodes plus modernes que celles permises par la norme, pour faire évoluer cet état de l'art. C'est ainsi qu'Airbus fut le premier constructeur à utiliser des méthodes de vérification formelle, et ces dernières seront intégrées à la future norme.

Plus qu'une vérification du logiciel proprement dit, la procédure de certification d'un système consiste donc avant tout à vérifier que l'industriel "maîtrise son sujet". Les organismes de certification tiennent à s'assurer que le développement du logiciel a été effectué de manière rigoureuse, en utilisant un enchaînement de tâches bien déterminé (selon le fameux cycle de conception en V). Surtout, ils veulent éviter que les industriels utilisent systématiquement chaque nouvelle technologie de développement qui voit le jour. En proposant une liste finie de méthodes et d'outils autorisés, les concepteurs ne peuvent pas utiliser des techniques non adaptées au monde du logiciel embarqué critique. Cela évite également que le nombre de lignes de code dans les calculateurs ne croisse trop rapidement et augmente de ce fait les risques de bugs. Il faut savoir que la certification d'un équipement est une procédure très coûteuse dont le montant est directement proportionnel au nombre de lignes de code (plusieurs dizaines d'euros en moyenne par ligne). D'où l'importance de bien maîtriser son programme et d'être capable d'en justifier chaque ligne, chaque variable, chaque opérateur logique. D'autant plus que si l'on vise un niveau de certification élevé (la norme DO-178B introduit cinq niveaux de criticité différents), il faudra proposer davantage de documentation, fournir davantage de preuves et effectuer des tests supplémentaires.

D'ailleurs, en raison de la complexité de la certification, nombreux sont les industriels qui préfèrent faire appel à un prestataire de service pour leur développement. En fonction de ses compétences, ce prestataire pourra prendre en charge tout ou partie du processus : développement, tests, validation, voire même jusqu'aux démarches de certification auprès de l'organisme approprié. Le recours à un prestataire est d'autant plus utile si l'industriel est confronté pour la première fois aux contraintes de la certification logicielle. D'après Thierry Dauty, responsable de l'activité tests logiciels chez *Sogeti High Tech*, les industriels délèguent de plus en plus de logiciels critiques à des sociétés de services. « *Pour l'équipementier ou le constructeur qui fait appel à nous, travailler au forfait présente un caractère rassurant (il n'y a pas de surprise lorsqu'arrive la facture)*, explique-t-il. *Mais ce n'est pas le seul avantage. Comme nous travaillons dans plusieurs secteurs simultané-* ➜

## DO-178C : un nouvel état de l'art en vue pour l'industrie aéronautique

La norme DO-178B appartient à cette catégorie de standards qui représentent un "état de l'art". Au moment de la définition du document, parmi les méthodes existantes ont été choisies celles qui garantissaient une sécurité fonctionnelle de haut niveau pour les systèmes aéronautiques. « *Le constat de base de la DO est qu'il est impossible d'affirmer qu'un code est fiable à 100 %*, explique Cyrille Comar, directeur général d'*AdaCore*. *Le principe consiste donc à mettre en place un certain nombre de méthodes et d'objectifs pour que le développement soit fait de la manière la plus rigoureuse possible.* » D'un commun accord, avionneurs, éditeurs et organismes de certification ont mis au point le processus de développement décrit dans le texte de la DO-178B qui respecte le cycle de développement en V, et utilise toutes les méthodes d'analyse de code disponibles à l'époque (analyse statique, test de couverture, etc.).

Aujourd'hui, la DO-178B est connue et utilisée par tous les industriels du secteur. Seulement, depuis 1992, date de la publication de cette norme, les technologies informatiques n'ont cessé d'évoluer. De nouvelles approches telles que le Model Based Design (conception orientée modèle) et le Model Based Testing (tests orientés modèle) remplacent de plus en plus systématiquement les développements classiques. Les avantages sont nombreux et ne sont plus à prouver (possibilité d'instancier des objets, de générer du code automatiquement, etc.). Il n'empêche que ces approches ne sont pas prises en compte dans la DO-178B. Du coup, l'industriel qui voudrait les utiliser s'expose à un travail titanesque, car il lui faudrait justifier auprès des autorités de certification que ces nouvelles méthodes fournissent du code suffisamment robuste.

Il en va de même avec les "méthodes formelles". Ces méthodes visent à prouver des propriétés non pas grâce à des tests, comme c'est le cas habituellement, mais grâce à des équations mathématiques. « *Les méthodes formelles sont vraiment bien adaptées pour des contrôles du type : vérifier qu'une fonction ne pourra pas aboutir à une division par zéro, vérifier qu'un calcul ne va pas entraîner un dépassement de capacité pour une variable, etc., et aussi apporter la preuve que le système ne pourra jamais se retrouver dans une situation incorrecte*, commente Marc Lalo, chef produits Polyspace chez *Mathworks*. *Même si elles sont plutôt adaptées au test de composants plutôt qu'au test de systèmes complets, les méthodes formelles offrent tout de même un niveau de confiance que les seuls tests ne permettent pas d'atteindre.* »

La vérification formelle est déjà intégrée à certains outils du marché (Polyspace de *MathWorks*, Scade d'*Esterel Technologies*, Spark d'*AdaCore*, Absynth ou encore Frama-C, l'outil réalisé par le CEA List et l'INRIA). Ces méthodes intéressent évidemment les industriels de l'aéronautique (une équation peut remplacer des milliers de tests). Pour preuve, *Airbus* les a utilisées pour la conception de l'A380. Un choix qui apporte donc un niveau de confiance supplémentaire, mais qui n'a rien changé du point de vue de la certification, car l'avionneur a tout de même dû procéder à tous les tests prévus par la DO.

Le besoin se faisait donc sentir de "remettre à jour" la DO-178B, pour lui ajouter les concepts du Model Based Design et de la preuve formelle (et pour prendre en compte les différents documents additionnels qui ont été publiés depuis 1992). Cela sera l'objet de la DO-178C, nouveau standard qui se superposera à l'actuel : chaque nouvelle activité prise en compte dans la version C retirera ou modifiera une activité prévue par la version B.

Le comité de rédaction de la DO-178C est uniquement composé de volontaires (un tiers d'industriels de l'aéronautique, un tiers d'organismes de certification et un tiers d'éditeurs de logiciels). Ce nouveau document de référence, actuellement en cours de finalisation, devrait être publié d'ici un à deux ans.

## Des kits de certification pour réduire les coûts

Pendant près d'une vingtaine d'années, les constructeurs aéronautiques ont maîtrisé l'intégralité de leur code. Chacun s'est doté de ses propres systèmes d'exploitation, environnements de développement et outils de tests. Mais au début des années 2000, les choses changent. Les constructeurs font de plus en plus appel à des sous-traitants qui ne connaissent pas forcément ces outils propriétaires. Parallèlement, il devient de moins en moins intéressant pour les constructeurs de conserver ces compétences en interne : les logiciels sont difficiles à maintenir, et dans le même temps de nombreux éditeurs spécialisés dans un domaine particulier ont fait leur apparition sur le marché. Dans ce contexte, pourquoi continuer à maintenir un système d'exploitation propriétaire lorsqu'un éditeur peut le faire mieux et pour moins cher ? La norme DO-178B impose aux industriels de conserver une certaine expertise de leur développement logiciel. S'ils veulent remplacer une étape manuelle par un outil, il est impératif que cet outil soit qualifié. Pour répondre à ce besoin, depuis quelques années, les éditeurs opérant dans l'aéronautique sont nombreux à proposer des outils "qualifiés" (on parle aussi de "kits de qualification" pour un outil, ou de kits de certification pour un composant logiciel). Et ils sont de plus en plus nombreux à disposer d'outils compatibles avec les degrés de criticité les plus élevés de la norme.

Un kit de qualification comprend de la documentation sur l'outil, expliquant son fonctionnement, ainsi que des cas de tests et des procédures pour le test des applications. L'intérêt est évident : en se présentant devant un organisme de certification avec un document fourni par l'éditeur, prouvant que son outil ne va pas rajouter d'erreurs dans le code, le constructeur s'affranchit d'une partie importante du travail de justification. Bien entendu, c'est désormais à l'éditeur (et non plus au constructeur) de se tenir informé des évolutions de la norme et de présenter aux autorités compétentes chaque nouvelle version de son logiciel.

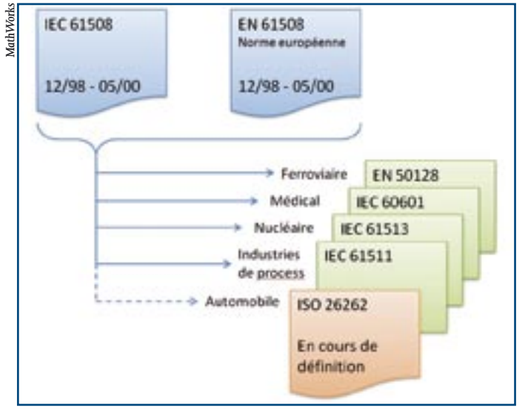
Passer du statut d'éditeur de logiciels "classiques" à celui d'éditeur de logiciels qualifiés ne s'est évidemment pas fait du jour au lendemain. Dans le domaine des systèmes d'exploitation temps réel, par exemple, les ingénieurs ont parfois dû se replonger dans leur code afin de l'alléger. « *Même s'il ne s'agit pas d'une certification à proprement parler (on certifie un avion ou un système complet, pas un composant logiciel seul), cette étape de qualification des outils n'est pas triviale, loin de là*, assure Olivier Charrier (*Wind River*). *Nous avons restreint les fonctionnalités de notre noyau de traitement temps réel, de sorte que le coût de la qualification diminue, mais aussi pour justifier d'un déterminisme du comportement du noyau. Nous apportons le minimum de modifications d'une année sur l'autre, afin de réutiliser les qualifications précédentes.* »

Aujourd'hui, la plupart des éditeurs d'outils de développement et de composants logiciels proposent ce type de kits. Ils sont disponibles pour les systèmes d'exploitation, les outils de développement et la plupart des outils de test. Tout le monde s'y met, à l'instar de Klaus Lambertz, cofondateur de la société *Verifysoft Technology*, qui constate que la demande est très forte : « *les deux tiers de nos clients industriels sont aujourd'hui confrontés à des problématiques de certification, qui sont le plus souvent imposées par leurs donneurs d'ordre. C'est la raison pour laquelle nous avons décidé de franchir le pas. Nous serons bientôt en mesure de proposer des outils de tests qualifiés pour les tests unitaires et les tests de couverture* ».

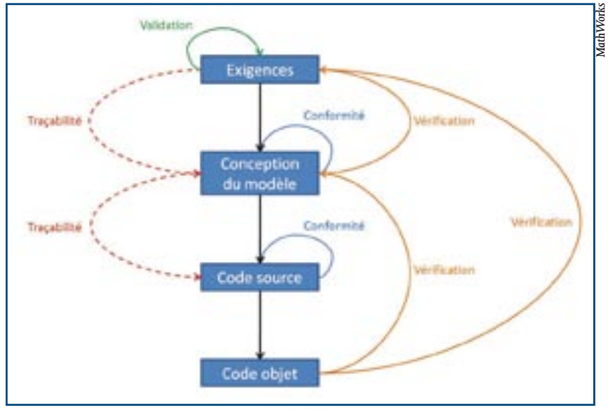
Nous l'avons vu, la norme DO-178B est avant tout une "philosophie". Elle impose notamment que le développement et la vérification d'un programme soient confiés à des équipes indépendantes (pour les plus hauts niveaux de criticité), et définit un certain nombre d'"activités" (ou "objectifs") obligatoires. Le terme activité désigne ici une étape indispensable dans le processus de développement, comme la traçabilité des exigences, la traçabilité des spécifications ou encore les tests logiciels (tests unitaires, de couverture et d'intégration). ➜

➜ ment, nous réutilisons certaines bonnes pratiques d'un secteur à l'autre. À cela s'ajoute une capacité de montée en charge importante : grâce au nombre d'ingénieurs disponibles, nous pouvons mobiliser de grandes équipes très rapidement afin de répondre à de vastes projets dans des délais très courts. »

Nous mettons ici le doigt sur une tendance forte du domaine : « *on observe un transfert de responsabilités des avionneurs vers leurs équipementiers, et des équipementiers vers les prestataires* », poursuit Thierry Dauty (*Sogeti High Tech*).




La norme IEC 61508, relative à la sécurité fonctionnelle des systèmes électriques, électroniques et électroniques programmables, sert de référence à la plupart des autres standards industriels.




Dans le processus de développement défini par la norme DO-178B, chaque étape fait l'objet d'une vérification par rapport à la précédente, afin que le code corresponde exactement aux exigences.

## Sécurité et sécurité fonctionnelle : deux concepts à ne pas confondre

Dans le monde de l'informatique embarquée, le concept de sécurité fonctionnelle (en anglais "safety") est bien distinct de celui de la sécurité ("security"). Si le premier désigne l'évaluation et la prévision des risques d'accidents, le second correspond à la protection contre les menaces extérieures. « *Pour expliquer cette différence, prenons l'exemple d'un bras de robot*, expose Olivier Charrier, ingénieur principal pour les activités Aérospatial et Défense chez *Wind River*. *Évaluer la sécurité fonctionnelle, c'est étudier les cas dans lesquels le bras peut heurter ou blesser un opérateur. Faire l'étude de la sécurité, en revanche, c'est s'assurer qu'il y a suffisamment de pare-feu, de dispositifs de contrôle d'accès ou d'appareils de cryptage pour éviter qu'une personne mal intentionnée ne prenne le contrôle du robot à distance.* »

Si les impératifs de sécurité fonctionnelle font référence à la norme IEC 61508 et ses dérivées (ainsi que DO-178B dans l'aéronautique), le standard de référence pour la sécurité s'appelle ISO 15408. Il est aussi connu sous le nom de "Common Criteria" (critères communs). Les techniques mises en œuvre pour la lutte contre la cybercriminalité étant récentes et coûteuses, il en va de même pour la certification proprement dite. « *On estime qu'une certification Common Criteria coûte environ vingt fois plus cher qu'une certification DO-178B*, poursuit Olivier Charrier (*Wind River*). *Pour donner un ordre d'idées, disons qu'une certification DO de niveau A (le plus élevé) revient aux alentours de 50 $ la ligne de code, tandis qu'une certification critères communs de niveau 6+ tourne plutôt autour de 1000 $ la ligne. Et encore, ce sont des prix pour du code bien conçu, lorsqu'il n'y a pas de retours en arrière à faire dans la conception…* »

En France, l'organisme en charge de la certification aux critères communs est l'ANSSI (Agence nationale de la sécurité des systèmes d'information). Bien sûr, tous les industriels n'ont pas l'obligation de se conformer aux critères communs, mais la certification est devenue systématique dans certains secteurs, comme celui de la carte à puce.

Enfin, les critères communs intéressent de plus en plus les constructeurs aéronautiques, surtout depuis que les contrôleurs aériens ont exprimé le souhait de pouvoir envoyer des données au pilote automatique depuis les tours de contrôle, afin d'aider les pilotes à atterrir en cas de besoin. À cela s'ajoute l'annonce récente du Parlement européen, qui souhaite uniformiser l'espace aérien. Un système informatique unique étant forcément propice à davantage d'attaques. À l'avenir donc, la simple sécurité fonctionnelle ne suffira pas. Il faudra impérativement aller vers des systèmes "safe and secure" intégrant les deux aspects de la sécurité.

## ➜ Des outils qualifiés tout au long du cycle en V

Pour chacune de ces activités, des éditeurs proposent des outils qualifiés sous la forme de kits d'aide à la certification. Ainsi, choisir aujourd'hui un outil non qualifié revient en quelque sorte à effectuer du développement "à la main" : il faut tout tracer, tout justifier, mettre en place ses jeux de tests, et cela peut être très long (deux ans en moyenne pour une application critique). En choisissant un outil qualifié, l'industriel peut automatiser une ou plusieurs de ces étapes. De par cette qualification, les organismes de certification valideront systématiquement le résultat fourni par cet outil.

Durant le cycle de développement, l'une des premières étapes consiste à tirer du cahier des charges un ensemble d'exigences, à partir desquelles on établit une liste de spécifications (la réponse attendue par le système lorsqu'il est soumis à une sollicitation particulière). Concernant la rédaction des exigences, rien n'oblige à employer un outil spécifique. Dans l'absolu, les industriels pourraient se contenter de documents Word ou Excel, comme c'est le cas dans les secteurs non critiques. Mais dans les faits, il s'avère que des outils du type Rational Doors d'*IBM*, Reqtify de *Geensoft* ou Quality Center de *HP Software* permettent de gagner beaucoup de temps. Sans leur aide, il faut vérifier manuellement que chaque exigence est bien prise en compte. Un travail colossal pour un avion complet !

L'éditeur américain *Visure* propose un outil d'ingénierie des exigences. D'après Didier Vidal, directeur général d'*ISIT* (qui distribue ce produit), « *Visure IRQA va plus loin que la seule gestion des exigences. En plus du suivi et de la mise à jour des exigences, cet outil trace les éventuelles dépendances entre exigences et favorise leur réutilisation d'un projet sur l'autre. En outre, il instaure un processus commun à toute l'entreprise pour s'assurer que tous les services ont compris les exigences et qu'elles sont correctement appliquées. Enfin, des connecteurs sont disponibles pour la plupart des outils de test logiciel du marché, afin que développeurs et testeurs bénéficient d'une interface unique.* »

Les outils de test logiciel n'échappent pas non plus à cette tendance. Parmi les outils de tests couramment utilisés dans l'aéronautique et qui bénéficient de kits de certification, on peut citer la suite Polyspace de *Mathworks*, la suite Testbed de *LDRA* (distribué en France par ISIT) ou encore Rational Test RealTime d'*IBM*.

« *Il est important de préciser que, dans le domaine du test, l'industriel a le choix d'utiliser un outil qualifié ou non*, explique Marc Lalo, chef produits Polyspace chez *Mathworks*. *Il faut dire que la norme DO-178B est relativement souple vis-à-vis des logiciels qui ne modifient pas le code (ils ne font que le contrôler, donc ils ne peuvent pas introduire d'erreurs).* » Ainsi, tous les outils courants de test logiciel pourront être utilisés dans le cadre d'une certification, du moment que leur comportement est validé (on doit "tester l'outil de test") et que leur utilisation est correctement définie et justifiée au sein du processus de développement. On pense aux outils de test unitaire (fournis par *Testwell*, *Parasoft* ou autre), aux outils de couverture de test (disponibles auprès de *Bullseye Testing Technology* ou d'*AdaCore*, par exemple), ou aux outils d'analyse statique (édités par *Coverity*, *Grammatech* ou *Klocwork*, entre autres). Notons tout de même qu'à côté des outils de test proprement dits, on trouve des logiciels faisant office d'"environnements de tests". C'est le cas de Quality Center (*HP Software*), qui propose d'automatiser les tests tout en conservant le lien entre chaque test et l'exigence à laquelle il correspond. Citons également Test Designer, de la société bisontine *Smartesting*. Contrairement aux autres logiciels qui créent des tests à partir du code, ce logiciel propose une génération automatique des cas de tests à partir d'une modélisation de type UML. « *Il ne s'agit pas d'un outil qualifié en tant que tel, car il n'évite pas la phase de justification, mais il accélère tout de même les procédures de certification*, assure Bruno Legeard, directeur technique de *Smartesting*. *La valeur ajoutée de Test Designer est de faciliter la réutilisation et la mise à jour des tests.* »

Une fois le programme spécifié et déve- ➜

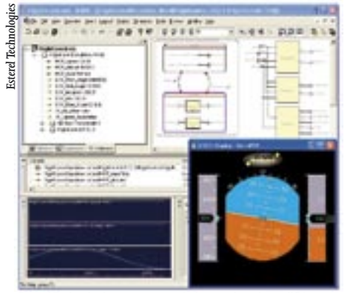
### Certification logicielle : quelques conseils avant de se lancer

Certifier un logiciel ne s'improvise pas. Les développeurs d'applications critiques sont devenus des experts par la force des choses, mais certains industriels d'autres secteurs peuvent aussi s'intéresser à la certification. Quelle que soit la raison (volonté d'améliorer la qualité de son code, souhait de se prémunir contre d'éventuelles actions en justice, ou encore critère imposé par le donneur d'ordre) la certification est une procédure complexe. Elle peut vite virer au cauchemar en cas de préparation insuffisante. Voici donc quelques conseils pour aider une entreprise qui doit certifier un programme pour la première fois :

- **Choisir un standard de référence.** En aucun cas on ne peut se permettre de respecter tous les standards. Il faut donc faire des choix. L'industriel qui choisit le standard IEC 61508 doit également sélectionner parmi les centaines d'activités différentes celles qui sont les plus adaptées à son métier. Attention également à choisir un standard compatible avec le pays dans lequel le produit sera commercialisé (les normes médicales sont notamment différentes entre l'Europe et les Etats-Unis).
- **Choisir un degré de certification adapté.** Chaque standard prévoit différents niveaux de certification selon la criticité de l'application. Passer à un niveau supérieur induit davantage de documentation, de justification et de tests. Pour cette étape, il est utile de se faire accompagner par un expert en safety afin de déterminer à quel niveau de criticité se conformer.
- **Sous-traiter certaines parties du processus.** Faire appel à un prestataire permet d'accélérer certaines étapes du développement. On évite surtout les mauvaises surprises : certes, la certification coûte cher, mais cela coûte encore plus cher de se la voir refusée lors du premier passage devant l'organisme de certification. Sur ce point, les consultants peuvent apporter leur expertise et mettre en évidence les architectures "douteuses" qui seront à coup sûr refusées par les certificateurs.
- **Utiliser des produits sur étagère.** Désormais, toutes les cartes électroniques ont des bases communes. Les industriels peuvent se tourner de plus en plus systématiquement vers des produits sur étagère dits COTS (Commercial Off-The-Shelf). Mais seuls les produits COTS "qualifiés" (ou intégrés à un kit de certification) apportent l'assurance que le produit présente la rigueur demandée par le certificateur. Sans cette qualification, l'industriel devra déployer d'importants efforts pour acquérir l'expertise nécessaire sur le produit en question.
- **Prendre en compte le facteur humain.** Nous l'avons vu, certifier un programme, c'est avant tout transmettre une confiance. Inutile donc de se présenter devant un organisme de certification sans une confiance absolue dans son produit et la manière dont il a été développé. Mais les personnes chargées de délivrer l'agrément final n'en restent pas moins des êtres humains. Il est donc impossible d'étudier l'intégralité d'un dossier de certification (avant l'utilisation des DVD, un dossier pour la certification d'un calculateur pouvait représenter 200 kg de documents papier). C'est pourquoi ils vont procéder à des vérifications par échantillonnage. Attention donc à ne pas laisser de zones floues dans la documentation (le certificateur, qui connaît très bien son secteur, va aller directement pointer sur ces points). Attention également à se tenir informé des autres problèmes du secteur. Si une fonction a mauvaise presse à un instant donné (on pense, par exemple, aux accidents liés aux régulateurs de vitesse), il faut s'attendre à ce que le certificateur pose des questions particulièrement poussées à propos de cette fonction.

### L'isolement des OS facilite la certification

Depuis les années 2000, afin de gagner du poids et de l'espace, les avionneurs regroupent de plus en plus de fonctions sur un même calculateur. Encore faut-il pouvoir justifier que la séparation entre ces fonctions est fiable, et ceci sans que les coûts de certification (déjà élevés) n'atteignent des sommes astronomiques. C'est le début de l'approche avionique modulaire intégrée, ou **IMA (Integrated Modular Avionics).** Le principe : regrouper plusieurs fonctions sur un même calculateur, et utiliser un OS qui garantit l'indépendance entre les applications. Cela facilite les relations avec la sous-traitance, car les équipementiers peuvent développer des fonctions sans se soucier de savoir si d'autres fonctions utiliseront le même calculateur. Il leur suffit de se confor-mer au standard ARINC 653 qui définit le comportement de l'OS et de ses différents services, ainsi que l'isolation temporelle (si une application envoie un message alors que ça n'est pas à son tour de parler, le message peut être interrompu et faire planter tout le système). L'approche IMA est actuellement employée dans l'aéronautique et va continuer d'être utilisée dans les avions de prochaine génération. En revanche, les constructeurs de ces nouveaux avions seront confrontés à des problématiques de sécurisation des communications. Ils envisagent des solutions comme **MILS (Multiple Independant Levels of Security)**. Ce concept a été lancé en 1980 et appliqué depuis 2005 par les militaires. Plutôt que d'avoir un seul niveau de sécurité par calculateur, ce dernier peut regrouper des fonctions de criticité différente. Les OS compatibles MILS instaurent différentes politiques de gestion d'accès et de sécurité (protection contre les attaques extérieures) sur une plate-forme matérielle commune.


Esterel Technologies

Esterel Technologies propose un générateur de code qualifié DO-178B niveau A. Une version de ce générateur de code est prévue spécialement pour le développement des IHM des cockpits.

loppé, il faudra le compiler pour l'exécuter sur la carte électronique définitive. Seul problème : à chaque carte correspond une version différente du compilateur. Pour un éditeur, il serait trop coûteux de faire qualifier toutes les versions de tous ses compilateurs. L'industriel sera donc obligé de prouver que celui qu'il a choisi n'introduit pas d'erreurs, et pour cela il fournira des résultats de tests à l'organisme de certification.

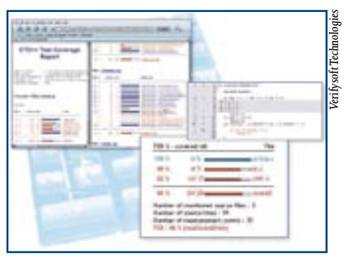
## Compilateurs et OS temps réel

Heureusement, certains éditeurs comme *AdaCore* fournissent des études de traçabilité pour leurs compilateurs. On retiendra également l'initiative de la société *Esterel Technologies*, éditeur de l'environnement de développement Scade, qui propose un kit de certification appelé CVK (pour Compiler Verification Kit). « *Nous savons que les industriels qui utilisent Scade adoptent le plus souvent des compilateurs tiers, choisis par eux ou par leur donneur d'ordre*, commente Luc Coyette, directeur technique chez *Esterel Technologies*. *Aussi, plutôt que de chercher à imposer l'utilisation d'un compilateur en particulier, nous proposons ce kit qui contient toutes les opérations de base que l'on peut créer avec un modèle Scade. L'industriel fait tourner ce "code de démonstration" sur sa cible et compare les résultats avec ceux obtenus dans l'environnement Scade. Ainsi, une fois devant l'organisme de certification, il peut indiquer qu'il a vérifié le fonctionnement de son compilateur avec ce kit qualifié, et les procédures seront simplifiées.* » Une approche unique sur le marché. Ajoutons qu'*Esterel Technologies* est le seul éditeur à proposer un générateur de code qualifié DO-178B de niveau A (le plus haut niveau de criticité). Si l'on sait que la génération automatique de code n'est pas une activité valable du point de vue de la DO-178B, on comprend que derrière ce kit de qualification se cache un travail colossal. Mais depuis qu'*Esterel* a su convaincre les organismes de certification, il n'est plus nécessaire d'effectuer des tests unitaires sur le code généré. Le gain de temps est évident. Rappelons tout de même que si Scade est largement utilisé dans l'embarqué critique depuis plus d'une dizaine d'années, ce langage formel est surtout adapté à la création de systèmes de contrôle-commande. Le plus souvent, on devra choisir un autre outil pour programmer le comportement des sous-systèmes. Chez *Mathworks*, on a choisi la philosophie inverse. L'éditeur américain, qui a un fort savoir-faire dans le domaine du test, préfère qualifier uniquement ses outils de vérification. Et ils sont nombreux : Simulink Verification & Validation, Simulink Model Advisor, System Test, Design Verifier ou encore Polyspace disposent tous déjà de kits pour faciliter la certification.

« *Au final, l'industriel a le choix : utiliser un générateur qualifié qui permet de se passer d'une partie des tests, ou utiliser un générateur classique et effectuer l'intégralité des vérifications de manière automatique. Nous pensons que cette dernière solution fait gagner plus de temps* », explique Daniel Martins, responsable des ingénieurs d'application chez *Mathworks*.

Dernière étape avant de finaliser l'application : l'intégration qui revêt une importance primordiale dans le cas des équipements certifiés. Là encore, l'industriel devra tout maîtriser et être capable de tout justifier. Les éditeurs spécialistes des environnements d'exécution proposent donc des versions qualifiées de leurs produits. C'est le cas d'*Atego* et d'*AdaCore*, qui fournissent tous deux des "runtimes" qualifiés DO-178B pour le langage Ada. « *Un runtime (ou environnement d'exécution) consiste en un noyau temps réel sur lequel tourne le programme*, explique Marc Richard-Foy, responsable des études avancées chez *Atego*. *En complément du langage Ada, nous réfléchissons actuellement à un runtime qualifié pour Java.* »

Qu'il soit sous forme de code compilé (on parle de code binaire) ou encapsulé dans un runtime, le programme est ensuite intégré à un système d'exploitation temps réel puis-


Verifysoft Technologies

Pour obtenir une certification, il faut apporter une preuve de la qualité des tests grâce à un outil de couverture de test. Il indique les parties déjà exécutées et les parties restant à tester.

qu'il s'agit d'applications critiques. Dans ce domaine, les éditeurs ont tous annoncé au cours des derniers mois des versions qualifiées de leurs RTOS (pour Real Time Operating System). Parmi eux, citons *Wind River* (avec VxWorks CERT), *Green Hills Software* (Integrity-178B), *LynuxWorks* (LynxOS-178), *Sysgo* (PikeOS) ou encore DDC-I (Deos DO-178B).

Toutefois, « *si l'offre d'outils pour l'aéronautique ne cesse de s'étoffer, à ce jour, il n'existe encore aucun logiciel capable de couvrir tout le spectre des activités de développement*, constate Cyrille Comar, directeur général d'*AdaCore*. *Les kits de certification font gagner du temps et de l'argent, mais il faudra porter une attention particulière à toutes les connexions entre logiciels : on doit éviter à tout prix d'introduire une erreur en passant d'un logiciel à l'autre.* »

## La certification pour les autres secteurs

L'aéronautique est certainement le secteur le plus avancé en matière de certification. Les procédures y sont bien connues et respectées depuis plusieurs décennies. Cependant, l'utilisation de produits sur étagère reste une pratique relativement récente (début des années 2000), et le fait que les éditeurs fournissent des kits de certification est vraiment un phénomène nouveau. Le développement de ces kits a réclamé d'importants efforts, qu'il faut tout naturellement rentabiliser. « *Nous cherchons à diversifier notre activité, pour déployer nos outils dans les secteurs du ferroviaire, de l'énergie et de l'industrie* », lance Luc Coyette (*Esterel Technologies*).

Mais passer d'une certification aéronautique à une certification industrielle n'est pas si évident, loin s'en faut. Les deux standards sont radicalement différents, comme l'explique Marc Lalo (*Mathworks*) : « *la norme DO-178B définit des obligations de moyens, tandis que la norme IEC 61508 définit des obligations d'objectifs (du moins pour la partie qui concerne le développement logiciel). Les moyens, ce sont par exemple l'indépendance imposée entre l'équipe de développeurs et l'équipe de testeurs, ou encore l'obligation de prouver que toutes les conditions d'appel d'une fonction ont été étudiées (activité appelée couverture MC/DC, pour Modified Decision/Condition Coverage). À l'inverse, l'IEC 61508 n'impose aucun processus ni aucune méthode d'analyse. L'obligation d'objectifs consiste à réaliser la fonction demandée, à savoir ne pas entraîner d'accident en cas de défaut. Dans l'industrie, le concept de sécurité impose de placer le système dans un état sécurisé en cas de défaut, tandis que dans l'aéronautique, il faut tout mettre en œuvre pour que l'avion ne s'arrête jamais.* »





## Le principe de l'IEC 61508

Le standard industriel IEC 61508 (ainsi que les normes qui en dérivent pour le ferroviaire, le nucléaire, l'automobile ou les industries de process) comporte une centaine de "tables". À chacune de ces tables correspond une manière différente de définir la sécurité fonctionnelle d'un équipement. L'industriel qui souhaite certifier son produit doit d'abord décider d'un processus de développement, et chaque étape de ce processus doit renvoyer à une table de la norme. Ce sera ensuite au TÜV (ou à tout autre organisme de certification travaillant pour l'industrie) de déterminer si la démarche suivie est valable, en fonction de la dangerosité de l'équipement.

Avec un nombre de processus de développement potentiellement infini, on comprend qu'il soit plus difficile pour les éditeurs de proposer des kits de certification. Difficile mais pas impossible : *Mathworks* dispose d'une offre IEC 61508 pour ses outils de test, *QNX* en propose pour son RTOS Neutrino comme *Wind River* avec VxWorks CERT, et *Esterel Technologies* fait de même pour son environnement de développement. La liste est encore longue, et au cours des années à venir, ce type de kits de certification à destination de l'industrie est appelé à se multiplier. Pour Didier Vidal, « *dans les secteurs rattachés à l'IEC 61508 et à ses dérivés, les constructeurs n'ont pas encore intégré les bonnes pratiques issues de l'aéronautique. Pourtant, cette rigueur dans les processus de développement est le seul moyen d'éviter les problèmes. Moins de 20 % de nos clients industriels utilisent un outil de gestion des exigences, alors que c'est un élément clé puisqu'il permet d'éviter près de 80 % des erreurs. On peut donc dire que la plupart des secteurs gagneraient beaucoup à utiliser des méthodes de développement inspirées de l'aéronautique* ». Sur ce point, d'ailleurs, les experts sont unanimes. Après l'aéronautique et le ferroviaire, le marché de la certification sera bientôt porté par deux nouveaux secteurs : l'industrie automobile et le secteur médical.

## L'automobile ne pourra échapper à la certification

L'ISO 26262, le standard dérivé de l'IEC 61508 pour l'industrie automobile, n'a pas encore été officiellement validé. Mais les constructeurs se tiennent prêts. Car avec l'accroissement du nombre de pannes logicielles (on se souvient des pannes de régulateurs de vitesse qui avaient fait grand bruit), la conformité à ce standard sera un jour ou l'autre impérative. Mais pour l'instant, il est économiquement impossible de la mettre en application : tous les composants d'une voiture se négocient au centime près. Difficile dans ces conditions d'investir des millions d'euros de recherche uniquement pour améliorer la sécurité de fonctionnement du logiciel embarqué.

À cela s'ajoute une autre tendance forte dans l'automobile : celle du "X-by-Wire". Les concepts "Steer-by-Wire" ou encore de "Brake-by-Wire" visent à remplacer les liaisons mécaniques du volant ou des freins par des algorithmes logiciels. « *Il est évident que nous y arriverons un jour*, assure Thierry Billoir (*Geensys*). *Mais avant cela, la conception des voitures devra être entièrement repensée car il faudra atteindre les mêmes niveaux de sécurité que dans un avion. Et même une fois qu'on aura atteint des niveaux de sécurité suffisants, il restera un autre problème à gérer : le changement fréquent du code embarqué dans l'automobile (pour chaque version de chaque modèle), qui représente un important travail de certification. Les constructeurs automobiles devront s'intéresser à de nouvelles méthodes de développement. On pense notamment aux méthodes formelles, qui ne sont pas prises en compte dans la version actuelle de la DO (il faudra attendre la version C) mais qui sont déjà intégrées aux normes issues de l'IEC 61508.* »

Le secteur médical devrait être, après l'automobile, la prochaine cible des éditeurs de logiciels. Certains ont d'ailleurs commencé : Klaus Lambertz (*Verifysoft Technologies*) nous confie qu'« *avec la crise économique qui vient de toucher l'automobile, nous avons encouragé nos commerciaux à démarcher les fabricants de matériel médical.* » Ces derniers sont, eux aussi, confrontés à une très forte augmentation de la proportion de logiciel dans leurs équipements. « *Non seulement les systèmes médicaux sont de plus en plus puissants (pour traiter des fichiers d'imagerie de plus en plus volumineux), mais on embarque désormais des processeurs 32 bits pour réguler le débit des seringues ou contrôler les pacemakers* », lance Stéphane Deruelle, directeur Europe du Sud chez *Wind River*. À cela s'ajoute le besoin de délimiter les responsabilités, car de plus en plus d'actions en justice sont engagées par les patients à l'encontre de leurs médecins.

Dans l'automobile comme dans le secteur médical, donc, les industriels doivent se tenir prêts à passer à un modèle économique dans lequel tous les produits devront être certifiés. « *Heureusement*, conclut Marc Richard-Foy (*Atego*), *ces industriels pourront profiter de toute l'expérience acquise dans l'aéronautique depuis près de trente ans. Il faut à tout prix les empêcher de chercher à réinventer la poudre.* »

**Frédéric Parisot**

### Des initiatives pour faciliter la certification

Les éditeurs de logiciels ne sont pas les seuls à réfléchir à des moyens de faciliter les processus de certification. Les membres des comités de standardisation sont chargés de trouver le bon compromis entre une utilisation systématique des nouveaux outils et un processus de développement figé et contraignant.

C'est notamment le rôle du groupe de travail DO-178C, qui se penche actuellement sur une révision de la norme DO-178B prenant en compte certaines nouvelles méthodes de développement.

Mais les industriels jouent, eux aussi, un rôle moteur, et n'hésitent pas à lancer des initiatives pour faciliter la certification. En France, le CG2E (Club des grandes entreprises de l'embarqué), créé à l'initiative du Syntec (Fédération des professionnels de l'ingénierie, des services informatiques, des études, du conseil et de la formation professionnelle), propose des services d'accompagnement. L'objectif : que l'industriel ne se retrouve pas seul face aux organismes de certification.

Dans le même esprit, l'initiative OpenDO lancée par *AdaCore* vise à répondre à deux problématiques : changer du code qui a déjà obtenu la certification, et mettre à jour un composant logiciel développé par une communauté (logiciel Open source). OpenDO réunit un écosystème d'industriels autour de différents sujets liés à la certification, afin que les bonnes pratiques développées par les grands groupes puissent servir à des entreprises de taille plus modeste.

Également à l'initiative de la société *AdaCore*, le projet Hi-Lite vise quant à lui le développement des méthodes formelles pour l'embarqué. Mené conjointement avec des chercheurs (le CEA-List et l'INRIA-ProVal) et des industriels (*Altran Praxis*, *Astrium Space Transportation* et *Thalès Communications*), ce projet aboutira à un environnement de développement spécialement adapté aux méthodes formelles. Il s'agit de rendre exécutable le programme regroupant les pré et postconditions de chaque fonction, de manière à faire apparaître les erreurs beaucoup plus rapidement.