Les véhicules autonomes seront-ils un jour sûrs?

WRITTEN BY

Christopher Giordano
Vice President UX/UI Technology
The DiSTI Corporation

Jim Carroll
Chief Technology Officer
Digica

Les véhicules autonomes seront-ils un jour sûrs ?

La sécurité des IHMs dans les véhicules autonomes (VA)

À première vue, il pourrait sembler que les interfaces utilisateurs (UI) des futurs véhicules autonomes n'auront pas d'exigences significatives en matière de sécurité fonctionnelle. En effet, le conducteur n'ayant plus ou quasiment plus à intervenir dans un véhicule autonome, l'importance et la criticité des IHM devraient diminuer. Les seules interfaces utilisateurs qui devraient rester seront probablement liées à l'info divertissement et de ce fait ne devraient pas être liées à une quelconque conformité de sécurité fonctionnelle.

Cependant, bien que les commandes traditionnelles liées à la conduite puissent devenir moins importantes, la nature des véhicules autonomes entraîne de nouvelles exigences supplémentaires pour leur contrôle. La normalisation des VA ne se fera pas du jour au lendemain non plus, pourtant les véhicules traditionnels toujours sous le contrôle d'un conducteur et les VA devront partager la route pendant la transition. Au cours de cette période de transition, les véhicules autonomes devront certainement conserver encore un certain niveau de commandes de conduite conventionnelles, commandes qui seront liées à des exigences de sécurité. Cet article traite de ce que pourront être ces futures interfaces utilisateurs, des défis ainsi que des solutions dans le développement de leur architecture logicielle et du processus de conception afin de répondre au besoin de sécurité fonctionnelle.

Quelle interface utilisateur pour les VA ?

Quel type d'interfaces utilisateurs et dans quelle catégorie des composants du véhicule elles doivent intervenir fait l'objet d'un débat ces derniers temps et dépend des sous-composants de l'interface utilisateur elle-même. Pour une IHM soumise à des exigences de sécurité fonctionnelle ASIL (niveau d'intégrité de la sécurité automobile), historiquement l'accent est mis sur les objets critiques intervenant généralement dans le tableau de bord (cluster instrument) du conducteur. Ces objets ou composants d'affichage sont directement liés à des éléments critiques du véhicules tels que les systèmes de contrôle des trajectoires, de traction, ou signalant des défauts du véhicule. Ces systèmes peuvent créer un environnement de défaillance « catastrophique », pouvant entrainer des pertes en vies humaines.

Jusqu'à récemment des domaines variés avaient des exigences d'interface utilisateur très différentes, en particulier en ce qui concerne la sécurité fonctionnelle. Avec l'apparition rapide des VA, les IHMs classées ASIL prennent une toute nouvelle signification pour ceux qui créent les exigences et le cas d'utilisation dans les différents composants des VA.

Info-divertissement

Par exemple, le système d'info-divertissement fait historiquement partie des systèmes non intégrés et dont le dysfonctionnement ne constitue généralement pas une menace pour la sécurité d'un véhicule en mouvement. Avec la convergence des sous-systèmes (un SoC gérant plusieurs écrans), l'info-divertissement commence à jouer davantage un rôle central dans le cockpit de conduite.

L'évolution matérielle où l'on passe d'un tableau de bord (cluster) et d'un affichage tête haute (HUD) séparés à une gestion par un système d'unité principale unique ayant la puissance et les capacités nécessaires pour piloter plusieurs écrans fait évoluer le champ d'action des exigences de sureté de fonctionnement. Bien qu'efficace du point de vue matériel, cette nouvelle architecture présente des défis uniques du point de vue du développeur logiciel, que nous abordons en détail dans cet article. Le matériel et la technologie d'affichage sont également arrivés au point où les automobiles grand public peuvent désormais utiliser des HUD.

Apporter un nouveau cycle d'IHM et d'architecture logicielle :

Quelles données afficher sur le pare-brise si ces données doivent être associées à des exigences ASIL et lesquelles ? Alors que les clusters et les HUD commencent à jouer un rôle de plus en plus important dans le système d'info-divertissement, une architecture fonctionnellement sûre associée à une culture de sécurité appropriée de l'entreprise devient d'une importance capitale pour le succès et la sécurité de la marque.

Télécommande

Les commandes de véhicules à distance sont également un bon candidat pour une interface utilisateur classée ASIL, mais ce que devrait être cette IHM n'est pas clair ou cohérent. Il existe plusieurs façons d'imaginer les fonctionnalités pilotables à distance:

» **Le conducteur à l'extérieur du véhicule déclenche un simple stationnement automatique à basse vitesse ou appelle des informations, généralement à partir d'un porte-clés ou d'un smartphone connecté.**

» **Le conducteur conduit dans la circulation et automatiquement prend en considération les événements tels que l'assistance au maintien de voie, le freinage automatique d'urgence ou le régulateur de vitesse adaptatif.**

Comme défini dans le premier cas ci-dessus, les capacités à distance sont généralement des commandes « tirer et oublier » au véhicule. Cependant, l'interface utilisateur doit toujours être claire, cohérente et fiable. Il contrôle le mouvement du véhicule, ce qui lui confère un certain niveau de classification ASIL, mais Comme défini dans le premier cas ci-dessus, ces capacités de contrôle à distance sont généralement des commandes « tire et oublie ». Cependant, l'interface utilisateur doit toujours être claire, cohérente et fiable.

Elle amène un contrôle du mouvement du véhicule, ce qui lui confère un certain niveau de classification ASIL, mais potentiellement pas sur l'IHM directement.

Dans le deuxième cas ci-dessus, ce type de capacités de contrôle à distance concernent d'avantage des ajustements mineurs du véhicule en mouvement. Le conducteur dans cette situation peut ne pas être en mesure ou avoir le temps de passer outre ses activités, par conséquent, une architecture sûre et fiable est également essentielle.

Reprise de contrôle d'urgence

Là où l'IHM est sans aucun doute d'une importance vitale dans les véhicules autonomes, c'est le concept de « dérogation d'urgence » (emergency Override), également appelé « événement de prise de contrôle ». Imaginez votre véhicule roulant sur l'autoroute à 120 km / h. Soudainement, un événement extérieur amène votre véhicule dans un nouvel état où il ne peut pas prendre de décision en raison d'un manque d'information. À ce stade, le véhicule déclenche un événement de prise de contrôle en urgence par le conducteur, lui donnant que quelques secondes pour connaître tous les états du véhicule afin qu'il prenne les bonnes décisions pouvant être vitales. Ces informations sont généralement transférées du véhicule au conducteur, de manière audible, visuelle voire les deux. Les études d'intégration multisensorielle montrent qu'une combinaison de données sonores et visuelles permet d'améliorer le temps de réaction moyen, d'où une réponse cognitive plus rapide.[1] Ainsi, afin d'améliorer ce transfert cognitif, disposer d'une interface utilisateur claire, concise et fiable devient crucial.

Il y a un débat de longue date sur le passage direct à l'autonomie totale sans passer par l'étape intermédiaire de systèmes semi-autonomes, qui supprimerait la nécessiter de reprise de conduite en urgence et donc la gestion des événements de reprises de contrôle. Les divers niveaux d'autonomie et leur signification seront abordés ultérieurement dans ce document.

1 - Rosemann, S., Wefel, I., Elis, V., & Fahle, M. (2017). Audio–visual interaction in visual motion detection: Synchrony versus Asynchrony. Journal of Optometry, 10(4), 242-251. doi:10.1016/j.optom.2016.12.003

Un long chemin vers les véhicules entièrement autonomes

Au fur et à mesure que les capacités des VA progressent, tout comme les définitions des exigences système des domaines présentés précédemment, les interfaces utilisateurs sont en perpétuelles évolution. Mais une chose est sûre : La sécurité fonctionnelle joue un rôle de plus en plus important dans l'architecture de l'interface utilisateur et du logiciel à mesure que les systèmes deviennent de plus en plus complexes. Simultanément, l'écart entre les véhicules à conduite traditionnelle et les véhicules autonomes se réduit. La période de transition entre aucune automatisation, une automatisation partielle et finalement une automatisation complète nécessite une définition claire.

En matière de conduite autonome, il existe cinq niveaux différents définis par la Society of Automotive Engineers (SAE) [2]

Comme mentionné précédemment, beaucoup pensent que passer directement vers l'autonomie totale est une approche plus sûre et que nous devrions passer directement au niveau 5. En théorie, cela peut être idéal car cela réduirait le fléau des accidents routiers, mais il y a d'autres considérations à prendre en compte telles que les coûts associés, le délai de mise sur le marché, les tests de tels systèmes et surtout le niveau d'acceptance du public face à ces nouvelles technologies.

En effet la volonté du grand public à adopter les VA reste mitigée et nous sommes actuellement dans une période de transition. Nous continuons à en apprendre davantage et à mettre en œuvre de nouveaux systèmes d'intelligence artificielle à même de prendre des décisions de conduite vitales et basés sur une quantité phénoménale de données acquises.

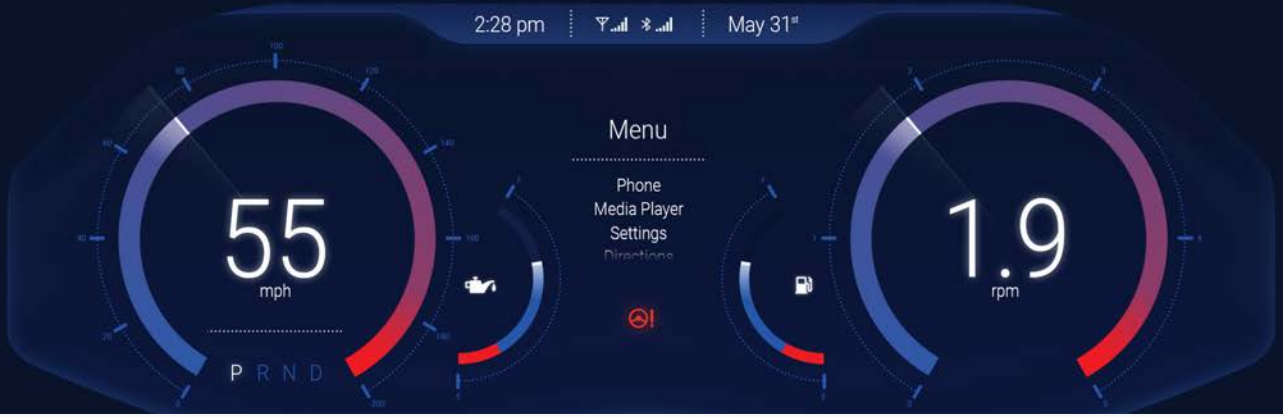
Ce fait montre que bien que n'étant pas entièrement hors de portée, nous avons encore un long chemin à parcourir avant d'atteindre le niveau 5 d'autonomie. Et Il y a en effet un bon nombre de facteurs à prendre en compte pour faire le saut vers le niveau 5, les principales questions étant:

- **La technologie utilisée est-elle sûre ?**
- **La culture de sécurité chez les OEMs et les Tiers 1 qui conçoivent et fabriquent les véhicules est-elle acceptable ?**
- **Comment les VA interagiront-ils avec les non-VA sur le réseau routier ?**
- **Devrait-il y avoir un certain niveau de contrôle humain sur ces types de véhicules et si oui, lequel ?**
- **Qui sera responsable des accidents avec un VA ?**
- **Qui sera l'autorité centralisée qui définira les normes qui seront exigées aux OEM sans faire augmenter drastiquement les coûts qui étoufferaient leur créativité ?**

Bien qu'il existe des avantages avérés en matière de sécurité, de mobilité et d'économie pour les véhicules autonomes, une exigence est évidente alors que nous continuons à nous rapprocher du niveau 5 : Il restera un certain niveau de contrôles humains ou de capacités d'intervention liés aux VA, du moins à court terme.

Et cela continuera probablement jusqu'à ce que nous prouvions que la technologie est sure, et que le public accepte que l'absence de contrôle humain amène à une conduite et un environnement beaucoup plus sûr.

2 - Lynberg, M. (2020, June 15). Automated Vehicles for Safety. Retrieved from https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety



Ce qui est important pour une interface utilisateur dans les VA

Pour un constructeur, les IHMs dans un véhicule représente en partie l'image de marque de l'entreprise. Pour certains l'apparence et la convivialité de l'interface utilisateur sont les principales différenciations d'une marque à une autre. En tant que tel, il y a toujours une compréhension implicite que l'une des principales exigences dans la conception d'une IHM soit que le rendu de celle-ci soit « beau », notion qui est subjectivement différente selon les concepteurs.

Pour des raisons de sécurité, l'interface utilisateur doit également être intuitive et simple afin de minimiser les distractions et les temps d'interaction. Quant au niveau de sécurité fonctionnelle attaché à ces UI, cela reste ouvert à l'interprétation par l'OEM et les fournisseurs. Il existe des directives de la NHTSA, du DOT, de la SAE et de la CE, mais pas de règles strictes de FuSa à respecter selon le niveau d'autonomie spécifique. Ces directives devraient changer et devenir plus rigoureuses à mesure que le niveau d'autonomie augmentera.

L'autonomie pure, une fois que nous y serons arrivés, aura son propre niveau de ce qui est important en termes de sûreté, cependant à court terme, pendant la période de transition, du point de vue de la sécurité, ce qui est essentiel pour l'interface utilisateur dans les véhicules autonomes est triple : simplicité, performance et sécurité.

- **Simplicité pour minimiser la charge cognitive et le transfert de données critiques vers l'humain lors des événements de prise de contrôle.**
- **Performances d'exécution pour garantir que le contenu critique s'affiche comme prévu au moment prévu.**
- **La sécurité fonctionnelle pour garantir que les composants critiques sont visibles avec précision et ce comme il a été défini durant la conception.**

Conduire dans le futur

Du point de vue de l'interface utilisateur, la réalisation de tous les points ci-dessus peut être gérée en utilisant des outils logiciels commerciaux éprouvés combinés à une architecture logicielle adaptée et avec une culture de sécurité à l'esprit. Le matériel et les logiciels continuent d'évoluer, deviennent plus performants et traitent plus de données plus rapidement. En tant que tel, il n'y aura jamais de processus de développement unique pour tous les systèmes. Afin d'assurer le changement et la pérennité de la technologie tout en permettant une adaptation rapide et efficace, la flexibilité deviendra la pierre angulaire du succès de tout OEM et fournisseur de niveau 1.

Par exemple, la tendance récente d'aller vers un calculateur unique pour piloter plusieurs types d'écrans n'était pas facilement possible il y a dix ans. Au fur et à mesure que le matériel est devenu plus petit, plus rapide, moins énergivore tout en pouvant transmettre d'énormes quantités de données à travers le bus, l'idée d'avoir un SoC capable de piloter plusieurs écrans, au lieu que chaque écran ait son propre SoC, est désormais accessible et financièrement viable.

Ce changement dans l'architecture matérielle a nécessité un changement dans l'architecture logicielle pour mélanger en un seul système des applications qui ne géraient pas traditionnellement des exigences ASIL (IVI) avec celles qui le devaient (Cluster & HUD). Ce nouveau concept de système unifié e-amène des défis uniques dans l'architecture logicielle du point de vue développeur. Mais il peut être fortement atténué en utilisant des outils commerciaux flexibles qui peuvent s'adapter sans mettre en danger l'architecture sous-jacente.

Outils Commerciaux

Il existe une perception selon laquelle les outils conçus pour répondre aux exigences de sécurité fonctionnelle sont obligatoirement plus chers que ceux qui ne le sont pas. Néanmoins adopter des outils utilisés sur des marchés tangentiels permet d'économiser les coûts d'ingénierie non récurrents immergés, réduisant ainsi les coûts de certification des logiciels pour les équipementiers automobiles.

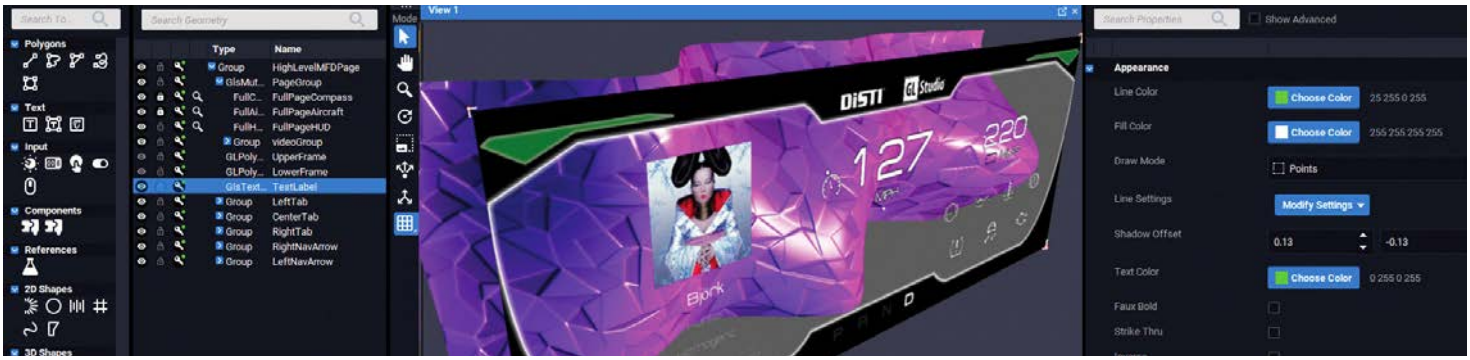
Tel que publié dans "Aviation Week & Space Technology", novembre 2017 par Rockwell Collins,

La réponse de Rockwell Collins est un système basé sur les risques, l'un des aspects des normes de l'industrie automobile. Le message est clair pour l'avionique et l'automobile - il y a beaucoup d'argent dépensé pour la sécurité. Comment les équipementiers peuvent-ils réduire les coûts, tout en respectant les normes de sécurité et sans étouffer l'innovation : En utilisant des outils commerciaux appropriés qui ont déjà supporté le plus gros des coûts et qui ont fait leurs preuves sur les marchés liés à la sécurité fonctionnelle.

> *"Auparavant, pour un nouveau système avionique, 75% de chaque dollar étaient dépensés en ingénierie et 25% en certification par des agences mondiales. Aujourd'hui, ce ratio a considérablement changé."* [3]

La certification des logiciels automobiles a défié les équipementiers au cours de la dernière décennie. Si les efforts de certification peuvent améliorer la sécurité automobile, le problème est que cela peut également freiner l'innovation, la créativité et augmenter les coûts de développement.

À une époque où les équipementiers automobiles cherchent à réduire les délais, les efforts à déployer pour la certification laissent à penser que cela augmente la bureaucratie, ce qui rallonge les délais de développement.



Les fabricants de semi-conducteurs ont reconnu l'importance de fournir des outils conformes FuSa pour avoir un avantage concurrentiel dans le secteur automobile. ARM propose un ensemble de fonctionnalités dans leur compilateur afin d'amener un niveau de sécurité dans le code. Comme ARM fournit également les IPs matérielles processeurs qui sont inclus dans de nombreux SoC, cela signifie que les fabricants de semi-conducteurs peuvent fournir à leur clients ce niveau de sécurité grâce au compilateur ARM avec un minimum d'effort supplémentaire. L'effet d'entraînement est que d'autres fournisseurs de SoC et d'IP concurrents de ARM doivent fournir des fonctionnalités similaires pour rester compétitifs dans ce secteur.

Tirer parti de l'expérience des outils utilisés dans d'autres domaines est une approche qui réduit considérablement les risques ainsi que les coûts tout en favorisant l'innovation. A titre d'exemple, la plateforme de conception d'IHM GL Studio de DiSTI a un pédigrée de plus de 20 ans dans l'avionique et le spatial et bon nombre d'équipements graphiques volent actuellement dans des avions, des engins spatiaux et des hélicoptères, ou encore est utilisée dans la conception de dispositifs médicaux critiques à travers le monde.

GL Studio a également été le premier outil de conception d'interfaces utilisateurs à être certifié selon la norme ISO 26262 ASIL D en avril 2015. GL Studio a également passé la pré-certification NQA-1 pour les installations nucléaires, un processus encore plus rigoureux que la certification DO-178C DAL A de la FAA pour l'avionique.4 Le coût de développement de ces outils a été amorti par l'étendue des marchés et des clients dans différentes industries ayant les mêmes considérations et besoins depuis plus de 30 ans, comme l'a fait l'automobile au cours de la dernière décennie. Ainsi, bénéficier de l'expérience et des outils existants pour des marchés tangentiels tels que l'avionique, secteur dans lequel la sécurité est primordiale et où les tests sont très rigoureux, devient une solution à faible risque et à faible coût.

Tout cela ensemble donne une architecture logicielle nettement plus fiable à une époque où les systèmes ADAS et les pilotes automatiques sont à l'horizon. Mais qu'est-ce qu'une IHM a à voir avec cela : Elle doit dorénavant prendre en charge les événements critiques et permettre la reprise de conduite d'urgence évoqué précédemment dans cet article.

3 - Statler, K. L. (2017, November 13). The World Needs Seamless Aviation Certification Standards. Aviation Week & Space Technology.
4 - HMI and UI Design Software: Embedded Target Systems. (2019, June 12). Retrieved August 28, 2020, from https://glstudio.disti.com/features/safetycritical/

Logiciels embarqués

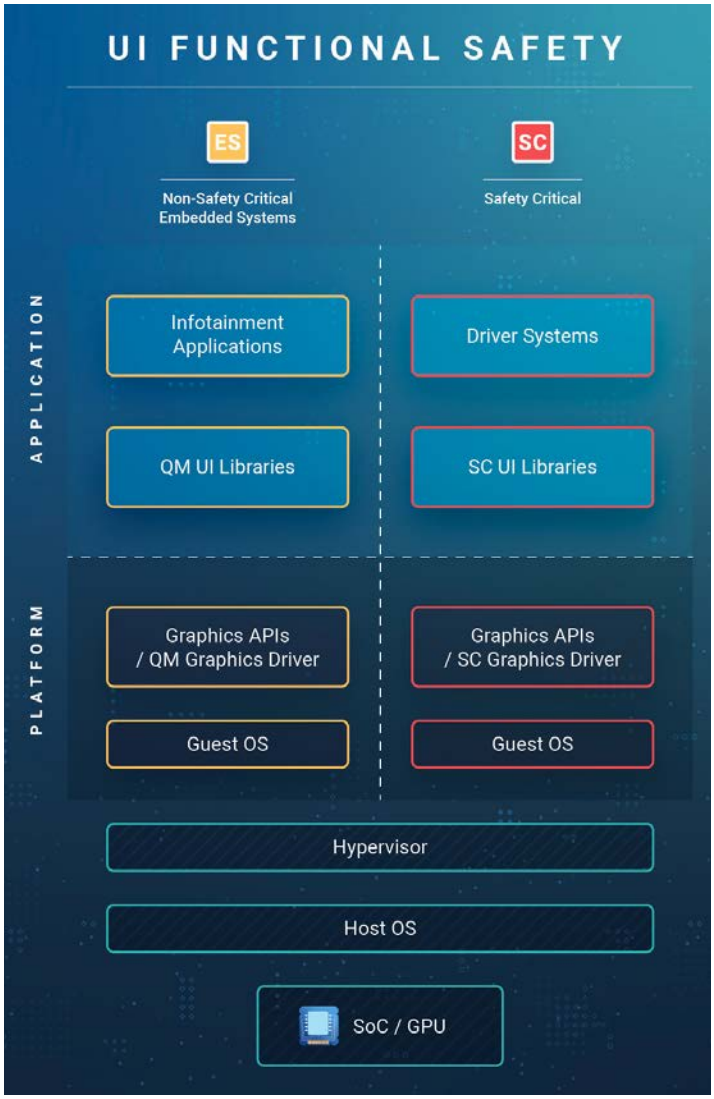
La sécurité fonctionnelle des interfaces utilisateurs dans les logiciels intervient dans deux domaines clés : Au niveau du firmware de la plate-forme elle-même qui exécute les parties applicatives critiques du système, et les applications avec lesquelles les utilisateurs interagissent.

Le firmware de la plate-forme comprend généralement de nombreux composants, y compris les pilotes des périphériques et autres drivers du BSP qui contrôlent directement les composants matériels sous-jacents, tels que les GPU, les protocoles, codecs et les couches d'abstraction fournissant des APIs cohérentes aux couches applicatives. Des organisations telles que Khronos spécifient des APIs standards à utiliser par les développeurs d'applications, telles que OpenGL, OpenGL ES et Vulkan. Des variantes sécuritaires (SC) de ces APIs sont également disponibles avec de nouvelles versions en cours de développement. Des drivers compatibles SC sont également disponibles pour les GPU pour prendre en charge ces APIs.

Lorsqu'il est nécessaire de partager les ressources matérielles entre les différentes fonctions du véhicule, la technique de virtualisation logicielle est généralement utilisée afin de réaliser cette séparation entre les fonctions. Dans un tel environnement, les pilotes hôtes et invités, ainsi que l'hyperviseur doivent également être conformes au niveau de sécurité
fonctionnelle requis.

Il existe également des capacités de virtualisation matérielle et de séparation GPU sur la même puce. Bien qu'il s'agisse d'un problème connexe, qui pourrait influencer l'architecture utilisée, la virtualisation matérielle sort du cadre
de l'architecture logicielle de cet article.

Comme le montre l'architecture de la figure de droite, les environnements logiciels virtualisés peuvent être complexes.



Cette approche peut compliquer le processus de résolution des défauts avant la production, allongeant le cycle de développement global.

Les outils et composants logiciels sécuritaires évoluent rapidement. A moyen terme, les hyperviseurs pourront devenir moins importants en tant que tels et les logiciels simplifieront la conception de systèmes critiques et sécuritaires.Cette approche réduira les délais de mise sur le marché, les coûts de développement et le prix des licences logicielles. Dans certains cas, il savère même que les outils SC sont au même coût que leurs équivalents non-SC. Dans ces cas, l'utilisation d'un outils SC permet même de simplifier l'architecture logicielle avec l'avantage supplémentaire d'une conformité SC complète sans frais supplémentaires.

Dans le contexte des IHMs automobiles, le logiciel applicatif est généralement l'interface utilisateur elle-même à travers laquelle les occupants interagissent avec le véhicule. Ce contexte inclut les fonctions du conducteur telles que le cluster, les commandes environnementales et l'info-divertissement. De telles applications peuvent être conçues pour être fonctionnellement sûres en utilisant des APIs sécuritaires et des outils de conception d'IHM certifiés tels que GL Studio.

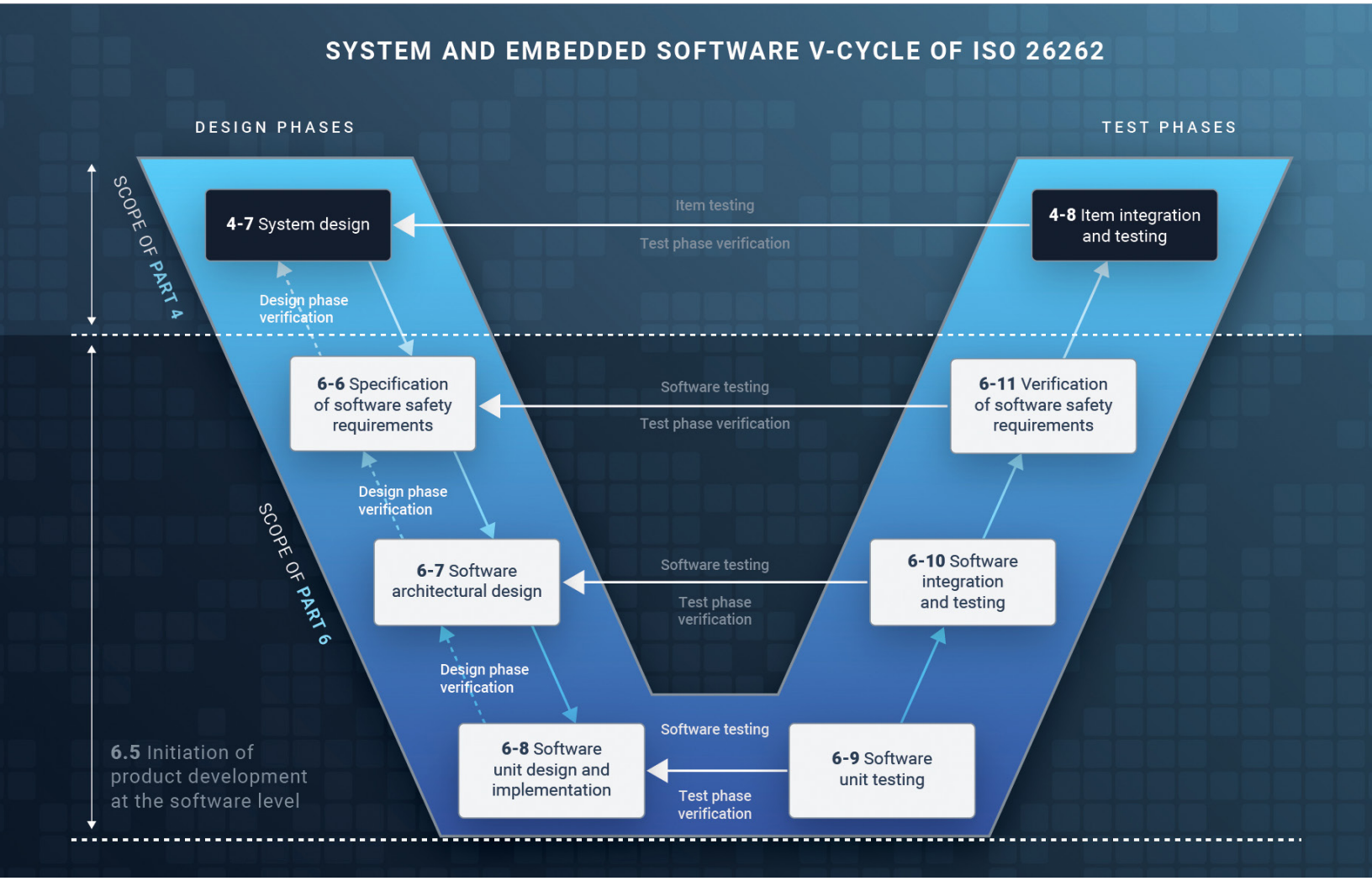
Séléctionner les bons composants logiciels sécuritaires appropriés pour son projet est une étape cruciale dans la phase de conception du produit, et ce, avant la programmation de tout code. La modification d'un outils ou d'une technologie de base au cours du cycle de développement entraînera une dette technique importante, des travaux d'intégration supplémentaires et peut compromettre la sécurité fonctionnelle du logiciel résultant.

Automotive Safety Integrity Level (ASIL)

La clé de la sécurité fonctionnelle est l'évaluation des risques. Des processus stricts sont imposés pendant le développement des véhicules afin de garantir le risque minimal pour les futurs utilisateurs finaux du véhicule. Un système de classification des risques, appelé ASIL (Automotive Safety Integrity Level), est fréquemment utilisé pour évaluer ces risques. Il existe quatre niveaux de risque, nommés ASIL A (le plus bas) à ASIL D (le plus élevé). Les dangers évalués ASIL D représentent des risques mortels, et ces développements sont soumis aux processus de développement les plus stricts.

En général, le principe à appliquer lorsque l'on travaille en ASIL est de travailler avec le plus haut niveau qui n'entraîne pas de coût supplémentaire mais qui permet de respecter les normes de l'industrie et la réglementation en vigueur. Une telle approche améliore la fiabilité du produit tout en permettant également une certaine flexibilité en cas de changement des exigences. Compte tenu des coûts de développement des logiciels dans des environnements critiques pour la sécurité, il est essentiel de prolonger leurs durées de vie et de réduire les coûts à long terme.

La classification ASIL est un sous-ensemble de la norme plus large, ISO 26262, qui fournit également des directives pour les processus de développement de produits dans l'industrie automobile. Le respect de ces mesures lors du développement des interfaces utilisateurs automobiles garantit la sécurité fonctionnelle.



Le processus est conçu pour garantir que les risques identifiés ASIL sont documentés, avec des solutions conçues, mises en œuvre et testées de manière appropriée. Un processus de développement dit « modèle en V » est décrit par la norme, qui met l'accent sur plusieurs caractéristiques clés :

- **Processus d'ingénierie, y compris la traçabilité par des exigences documentées, la conception et la mise en œuvre.**
- **Vérification structurée des résultats de chaque phase du développement, par des processus pairs dans le cycle de test.**
- **Focalisation spécifique sur les exigences de sécurité du système, en abordant les risques identifiés par ASIL.**

Le modèle en V (et ses variantes) est mature et couramment déployé dans les projets de développement de logiciels embarqués. Son utilisation s'applique également à l'ensemble des composants logiciels applicatifs automobiles, y compris les applications d'info-divertissement, les drivers, les bibliothèques IHM et les pilotes graphiques. **Tous les composants logiciels de l'interface utilisateur doivent être développés conformément à ces normes pour être conformes SC.**

GPU

Les interfaces utilisateur modernes sont incroyablement complexes et imprègnent tous les recoins d'un système d'exploitation moderne, dans tous les contextes déployés. L'interface utilisateur est au premier plan de l'expérience utilisateur à partir du moment où celui-ci entre dans le véhicule, de l'écran de démarrage de la marque du véhicule au système de menus et jusqu'à l'application de navigation. En raison de la complexité et de l'importance de l'expérience utilisateur, les performances sont essentielles. Toutes les interfaces utilisateurs modernes sont construites sur du matériel comprenant un GPU afin que les fonctions graphiques puissent être déchargées du processeur central.

La complexité du GPU est telle que les fabricants de SoC ne développent pas leur propre GPU mais intègrent des GPU tierces-parties dans leurs SoC. Imagination Technologies et ARM sont les principaux fournisseurs d'IP GPU « sous licence ». D'autres sociétés de SoC conçoivent leurs propres GPU qui sont intégrés à leurs processeurs applicatifs. Cette approche est adoptée par des sociétés telles que Nvidia, Qualcomm et Intel.

L'industrie a créé des organismes tels que Khronos pour définir des API standards pour la programmation graphique afin de simplifier l'intégration logicielle. OpenGL a été spécifié pour la première fois pour les PC de bureau il y a près de trois décennies. Il a subi de nombreuses révisions au fur et à mesure que le matériel a évolué et a engendré plusieurs API associées telles qu'OpenGL ES, à utiliser dans des contextes embarqués.

Des variantes sécuritaires de ces deux API OpenGL et OpenGL ES ainsi que les drivers pour les GPUs existants sont également disponibles auprès de sociétés telles que CoreAVI. Microsoft, de son côté a spécifié une API concurrente, Direct3D mais bien que Windows soit couramment utilisé dans les systèmes d'info-divertissement, la plupart de ces systèmes sont basés sur Linux et utilisent des API Khronos. Apple définit également une API concurrente appelée Metal. Actuellement, il n'y a pas de variante sécuritaire pour Direct3D ni pour Metal et à ce jour Métal n'a pas encore été déployé dans un système automobile commercial.

Comme les GPU sont également utilisés pour des calculs plus génériques, Khronos spécifie également une API de calcul, OpenCL et ils ont développé une alternative à leur ensemble d'API, appelée Vulkan. Cette méthode unifie les API graphiques et de calcul. Vulkan pourrait remplacer à termes les APIs actuelles, mais ce n'est en aucun cas une certitude - le modèle de programmation étant nettement plus compliqué que ses prédécesseurs. Pour unifier les graphiques et les calculs dans une seule API, une grande partie de la logique précédemment encapsulée dans le pilote GPU doit maintenant être implémentée par le développeur de l'application ou le fournisseur d'outils et de bibliothèques d'interface utilisateur. Bien que cela permette une plus grande différenciation des fonctionnalités pour les fournisseurs d'outils, cela introduit une complexité supplémentaire significative liée au matériel dans les composants qui ont jusqu'à présent été extraits des dépendances matérielles. En 2019, Khronos a commencé à travailler sur une variante critique sécuritaire de l'API Vulkan.

Autres considérations

Les équipementiers automobiles s'efforcent continuellement de réduire le coût de la nomenclature du véhicule. Cette approche peut conduire à la suppression de certains SoC, ce qui implique le partage de processeurs physiques entre les fonctions du véhicule. Comme indiqué précédemment dans l'article, il existe également des capacités de virtualisation GPU, qui seraient liées à cette fonctionnalité. Lorsque des fonctions critiques et non critiques pour la sécurité coexistent sur un seul processeur, il est techniquement plus simple d'utiliser une seule librairie logicielle graphique/IHM sécuritaire pour réaliser toutes les fonctions hébergées sur ce processeur.

Historiquement, les systèmes embarqués étaient basés sur des SoC SKUs standards. Les SoC sont des devices très complexes, et donc relativement chers. Pour des applications spécifiques, il est possible de concevoir un système en utilisant des composants moins coûteux, comme des FPGA ou des microcontrôleurs par exemple, tout en satisfaisant aux exigences d'une interface utilisateur de haute qualité et de conformité SC. Plusieurs éditeurs de logiciels (ISV) fournissent des logiciels de rendu, supprimant ainsi le besoin d'un GPU et les complexités de son partage entre les systèmes d'exploitation. Un GPU est de loin la plus grosse unité de calcul intégrée dans un SoC, donc le supprimer entraîne également une empreinte plus petite du SoC. Au moment de l'introduction des API telles qu'OpenGL ES, les moteurs de rendu logiciels ne fonctionnaient pas assez bien pour être inclus dans un environnement SC automobile. Aujourd'hui, leurs performances se sont nettement améliorées. Les possibilités des FPGA, y compris les GPU, ajoutent une dimension supplémentaire aux décisions architecturales.

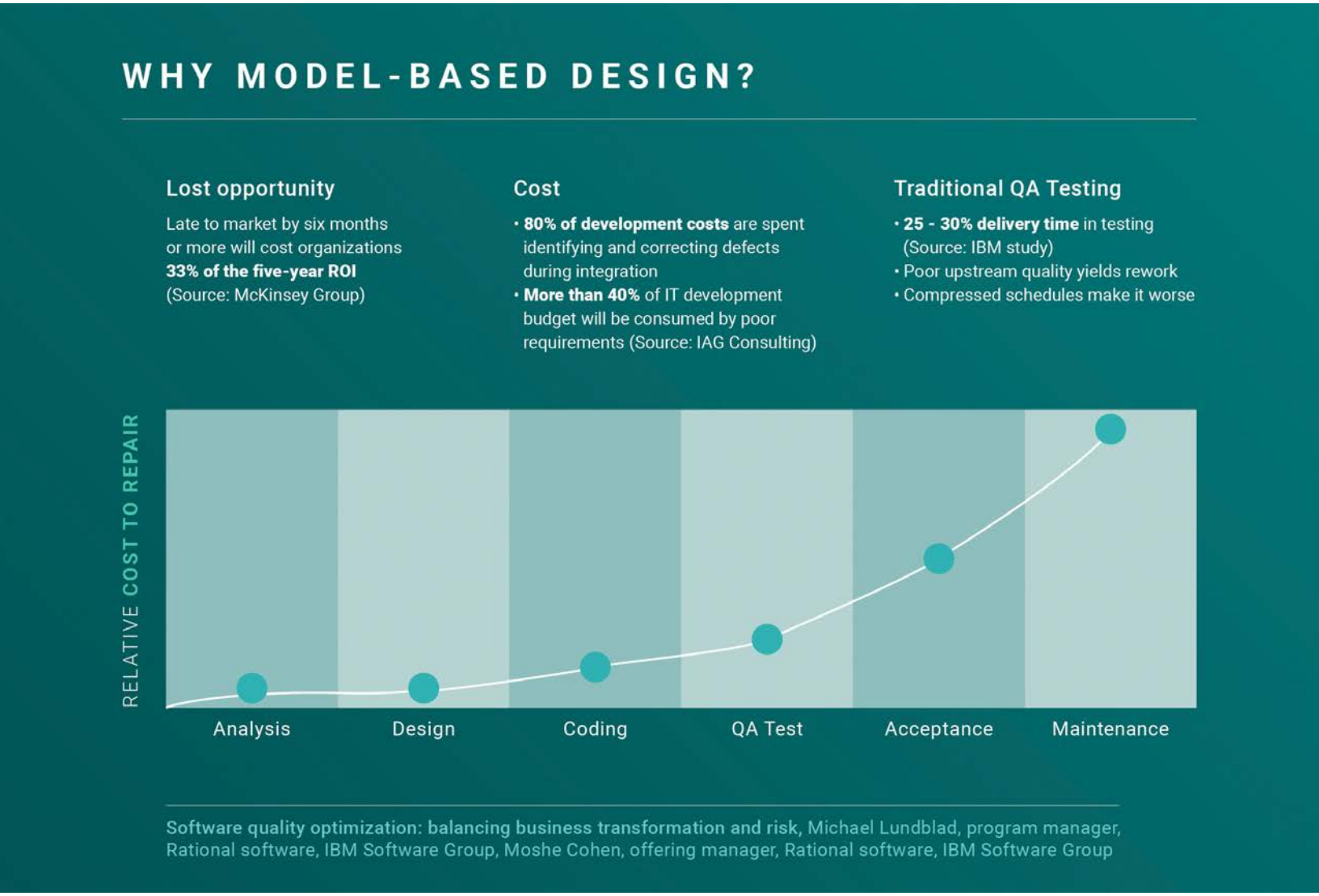
Les microcontrôleurs (MCU) sont généralement utilisés pour les applications « plus petites », qui bien souvent, n'incluent pas d'interfaces utilisateurs. Ils sont déjà largement utilisés, dans le contexte automobile, comme base des calculateurs pour les composants SC tels que les systèmes de freinage. Comme les FPGA, les microcontrôleurs ont également augmenté leur puissance de traitement au point que les performances des moteurs de rendu logiciels actuels sont aujourd'hui suffisantes pour être utilisés dans une plus large gamme d'applications embarquées. L'intérêt des microcontrôleurs est que leur consommation électrique est beaucoup plus faible que les SoC ou les FPGA et cela permet de réduire les coûts.

Si réduire les coûts peut être possible en sélectionnant le matériel le plus adapté, il est néanmoins essentiel de tenir compte de l'impact des coûts connexes tels que le coût de développement du logiciel ou des licences des logiciels tiers.

CONCLUSION

Alors que nous nous rapprochons de l'avènement des véhicules autonomes de niveau 5, il est évident que cela doit être accompli en toute sécurité. Pour les nombreuses raisons présentées dans cet article, le continuum des véhicules autonomes se rétrécit lentement sur la voie à suivre. Néanmoins même s'il est clair que nous y parviendrons, l'automatisation complète de niveau 5 n'est pas dans notre avenir immédiat. Il y a trop de facteurs impliqués pour un passage rapide et radical directement au niveau 5.
Bien qu'il y ait de nombreux avantages à tirer de l'autonomie pure, l'industrie automobile doit surmonter des problèmes d'infrastructure, des problèmes socio-économiques, les émotions humaines et, surtout des défis technologiques importants.

Alors que nous devons continuer à progresser, cet article met l'accent principalement sur la nécessité de le faire d'une manière sûre et économiquement viable. Comme mentionné dans une étude bien connue sur la conception basée sur les modèles, « 80% des coûts de développement sont consacrés à l'identification et à la correction des défauts pendant la phase d'intégration » d'un projet logiciel5.



Même si ce n'est pas un principe généralement bien accepté en raison des coûts initiaux, la clé du succès est d'investir dans la bonne architecture matérielle et logicielle ainsi que dans un flux de travail formalisé dès le départ. En fin de compte, la flexibilité requise pour pérenniser la conception de votre logiciel et s'adapter aux nouvelles technologies et architectures a un coût initial.
Mais comme le montre l'étude ci-dessus, si votre projet est correctement mis en œuvre dès le départ, au final, cette approche permet de réduire fortement les efforts et les coûts sur la durée de vie de votre projet.

L'objectif principal du développement d'un logiciel IHM embarqué doit être la sécurité. Au-delà de cela, l'équilibre entre les coûts, l'apparence et les performances de l'interface utilisateur peut être trouvé en sélectionnant l'architecture la plus appropriée mais également en tirant parti des outils COTS existants prenant en charge nativement les aspects sécuritaires permettant ainsi de ne pas étouffer l'innovation tout en vous assurant de ne pas dépasser votre budget.

5 - Lundblad, M. & Cohen, M. (march 2009). Software Quality Optimization: Balancing business transformation and risk. IBM Software Group

A propos des auteurs

Christopher Giordano

Vice President UX/UI Technology

The DiSTI Corporation

Depuis 1997, Chris s'est concentré sur le développement de logiciels d'interface utilisateur et d'IHM à partir de la marine américaine et de l'Université de Floride centrale. Chris travaille chez DiSTI depuis 1999 en tant qu'ingénieur principal et responsable de programmes pour plus de 60 programmes différents, et finalement chef de produit pour tous les outils de développement d'interface utilisateur de DiSTI. Chris a géré les programmes IHM / UI de DiSTI pour Boeing, Hyundai, Jaguar Land Rover, Lockheed, la NASA, Nissan Motors, Northrop Grumman et The Space Ship Company, pour n'en nommer que quelques-uns. Il est actuellement vice-président de la technologie UX / UI de DiSTI. Il a géré avec succès les activités UX / UI de DiSTI pendant plus d'une décennie, développant une position de leader mondial en tant qu'experts en IHM / UI et en sécurité fonctionnelle. Chris est titulaire d'un diplôme en finance de l'UNCW et en génie informatique de l'UCF avec mention.

Jim Carroll

Chief Technology Officer

Digica

Jim Carroll est un directeur technique très expérimenté avec une formation en systèmes embarqués. En 25 ans, Jim a conçu et développé des solutions logicielles à grande échelle atteignant des millions d'utilisateurs travaillant en tant qu'ingénieur logiciel, architecte technique et CTO. La majorité du travail de Jim a été dans le secteur des semi-conducteurs, travaillant avec des sociétés telles qu'Intel, Arm et Imagination Technologies. Il a également contribué au groupe de normalisation des logiciels, Khronos.