



Le projet AEE et la description UML des architectures embarquées pour l'automobile.

J.P. Elloy⁽¹⁾, **F. Simonot**⁽²⁾, **Y. Trinquet**⁽¹⁾

(1) : IRCCyN - ECN, UMR 6597 – Nantes –

(2) : LORIA - INPL, UMR 7503 – Nancy –

Actuellement l'électronique embarquée dans les véhicules croît très rapidement en nombre de fonctionnalités et ce dans tous les segments de véhicules. Cette croissance est due à l'apparition de dispositifs innovants (telle l'injection électronique), à la fourniture de services nouveaux à l'utilisateur (services de confort et de sécurité), au remplacement de fonctions électro-hydraulico-mécaniques par leur équivalent électronique. Pour permettre à l'industrie automobile française et européenne de rester concurrentielle dans ce secteur, le Ministère de l'Industrie a décidé de soutenir le projet AEE (Architecture Electronique Embarquée, sept. 98 – déc. 01) qui a pour but de définir une ossature logicielle unifiée des architectures embarquées dans les transports, d'assortir cette ossature d'outils de conception et d'analyse d'architecture, et de faciliter la mise en place de processus multi-partenaires pour développer ces architectures. Le projet AEE associe les constructeurs et équipementiers français du domaine ainsi que des universitaires du « temps réel ».

La nouveauté conceptuelle essentielle des nouvelles fonctionnalités automobile est que leur réalisation requiert de plus en plus la coopération de plusieurs fonctionnalités (ainsi l'ESP qui coordonne l'ABS, l'ASR et le contrôle d'inertie), ou bien utilise des informations délivrées par d'autres services (ainsi l'essuyage qui utilise la vitesse véhicule pour régler la cadence de balayage). A cela s'ajoute le besoin de « rentabiliser » les architectures déployées en optimisant l'exploitation des processeurs et des réseaux (rentabilisation économique, mais aussi en termes de performances temporelles ainsi que de sûreté de fonctionnement). L'intégrateur est désormais en face de contraintes nouvelles car il doit se plier au développement (partiellement) conjoint de différentes fonctionnalités, en effectuer une implantation optimisée sous contraintes dans une architecture matérielle répartie, et s'appuyer pour cela sur un processus de développement coopératif qui assure une progression validée du projet entre constructeurs et équipementiers tout au long de ses itérations et modifications, et délimite les responsabilités juridiques de chacun.

Les objectifs du projet AEE sont donc autant techniques qu'économiques : réutilisation des logiciels, portabilité des sources, indépendance de la conception fonctionnelle vis-à-vis de la distribution matérielle, rationalisation des calculateurs et réseaux, validation des exigences de bout en bout, vérifications comportementales et temporelles, sûreté de fonctionnement, simulations, tests, etc.

Le résultat d'AEE est de proposer une solution qui s'appuie sur un langage de description d'architecture qui a été défini spécifiquement : le langage AIL (Architecture Implementation Language). AIL définit le « template » générique de l'éventail des entités introduites au cours du développement d'une architecture complète. Ainsi, avec AIL, une architecture est décrite comme un ensemble de composants interdépendants qui sont des instances des classes d'un modèle générique. Ces classes, éprouvées sur démonstrateurs, définissent la plupart des caractéristiques fonctionnelles, matérielles, temporelles, et de SdF nécessaires aux analyses métiers de l'architecture globale et d'architectures partielles embarquées dans l'automobile.

Depuis la spécification de ses exigences jusqu'à sa mise en œuvre, une architecture embarquée se décrit en AIL en suivant cinq niveaux d'abstraction, chaque niveau modélisant un état d'avancement différent ou un aspect particulier de cette architecture. Le niveau le plus élevé est celui du « **projet véhicule** » dans lequel des classes AIL permettent de modéliser les services que l'architecture doit délivrer (anti-blocage de roues, régulation de vitesse, conditionnement d'air...) ainsi que leurs variantes (climatisation automatique ou manuelle, par exemple) et décrire les versions de véhicules qui en dérivent avec leurs prestations associées.

Le niveau « fonctionnel », qui succède au précédent, spécifie les composants fonctionnels de l'architecture, les raffine jusqu'à leurs constituants élémentaires et spécifie de même les interactions entre ces constituants. Ce niveau est indépendant de l'implémentation et donc de l'allocation des fonctions sur les calculateurs. L'**architecture fonctionnelle** qui est le graphe des fonctions de ce niveau et de leurs interactions constitue le pivot de la flexibilité, de la réutilisation et de la portabilité des architectures embarquées.

La mise en œuvre des fonctions s'appuie sur les classes des niveaux « logiciel » et « matériel ». Le niveau « logiciel » comprend les composants logiciels libres (c'est-à-dire dont le contenu algorithmique est indépendant du support d'exécution) et les pilotes logiciels des capteurs et actionneurs. A ce stade, AIL permet ainsi de construire une **architecture logicielle** exprimant les relations entre ces éléments logiciels, les organes d'instrumentation et les flots de données échangés entre ces constituants. Le niveau « matériel » est celui du support d'exécution : calculateurs, réseaux, liaisons série. Ces composants sont modélisés jusqu'à leur connectique, et aussi par leurs caractéristiques fonctionnelles et technologiques (consommation, par exemple) ; leur description est accompagnée de celle de leurs logiciels de gestion (drivers, exécutif, protocoles de communication...). L'**architecture matérielle** décrit les liens entre tous ces matériels et logiciels.

Au dernier niveau l'architecte obtient les tâches temps réel dérivées de son architecture logicielle, leur localisation dans l'architecture répartie, leur paramétrage en fonction du support d'exécution utilisé, et les trames réseau dérivées des données échangées entre fonctions distantes. L'**architecture opérationnelle** qui décrit cette organisation est obtenue par projection de l'architecture logicielle sur l'architecture matérielle, ce qui inclut (pour les logiciels libres) un calcul du placement des tâches logiques sur les calculateurs.

La présentation décrira les classes AIL essentielles du langage et leurs organisations dans les différentes architectures AIL. On montrera quels sont les jalons de développement et du processus coopératif qui sont attendus à chaque niveau, et le rôle des attributs des classes dans ces jalons. On évoquera enfin les aspects liés à l'échange d'informations entre partenaires ainsi que la structure générique d'un atelier de développement prenant en compte les exigences spécifiques de chaque acteur et ouvert vers différents outils de validation et d'analyse.