Les logiciels embarqués assurant les services de communication dans l'automobile - Contribution au Livrable 1.2 du projet SCARLET

Xavier Grandmougin, Françoise Simonot-Lion

Les logiciels embarqués assurant les services de communication dans l'automobile

Contribution au Livrable 1.2 du projet SCARLET

Septembre 2007

Auteurs : Xavier Grandmougin (LORIA, INRIA), Françoise Simonot-Lion (LORIA, INPL)

1.1 Objet d'étude : les logiciels embarqués assurant les services de communication dans l'automobile

Nous nous intéressons à la configuration des services de communication qui sont intégrés dans les logiciels ; ces services assurent les échanges entre composants applicatifs et ce de manière transparente vis-à-vis de la distribution de ces composants sur une plate-forme distribuée ; les composants logiciels applicatifs émettent et consomment des signaux tandis que sur le système de communication sont transmises des trames selon une politique induite par le protocole d'accès au médium (MAC). Le problème consiste à transformer un ensemble de signaux échangés entre composants applicatifs, selon une dynamique spécifiée au niveau applicatif, en une configuration de trames les transportant selon le protocole MAC. Ceci passe, entre autres, par des opérations de :

- *Fragmentation*, à l'émission, des signaux trop longs pour être contenus dans une trame et *défragmentation* correspondante lors de la réception ;

- *Frame-Packing*, à l'émission, c'est-à-dire insertion dans une trame unique de plusieurs signaux ou fragments de signaux, de taille inférieure à la longueur maximale de la trame, émis par des composants applicatifs différents sur le même ECU et *frame-unpacking* correspondant lors de la réception ;

- *Gestion des émissions, réceptions de trames.*

Le but de cette étude est la génération automatique de la partie de l'intergiciel qui assure, entre autres, ces fonctions selon les objectifs suivants :

- *le respect des contraintes de temps spécifiées sur les composants applicatifs* (respect de bornes sur le temps de réponse de composants, respect de bornes sur la gigue de fin d'exécution de composants périodiques, respect de bornes sur les temps de réponse de bout en bout sur une chaîne de composants s'exécutant sous des contraintes de précédence, etc.) ;

- *le respect des contraintes de temps spécifiées sur les signaux (informations élémentaires) échangés entre composants applicatifs* ; les propriétés sur les signaux sont cohérentes et sont parfois déduites des propriétés citées dans l'item précédent ; il s'agit par exemple de propriétés de fraîcheur d'un signal imposées par les composants applicatifs le consommant, de propriétés sur la gigue sur la fin de transmission pour un signal émis périodiquement, etc.

- *l'optimisation de certains paramètres* comme la minimisation de temps de réponse de bout en bout entre composants applicatifs, la minimisation de la durée de transmission de signaux non soumis à des contraintes de temps, la minimisation de bande passante, etc.

Certaines des propriétés citées ci-dessus se vérifient, en particulier, par la spécification d'un *ordonnancement faisable des tâches* qui réalisent ces composants et qui sont locales à chaque ECU (Electronic Control Unit) et par la préservation de cette faisabilité par les tâches de l'intergiciel qui seront générées. Lorsque les signaux sont échangés localement, les travaux se ramènent à l'étude des tâches vues précédemment. Lorsqu'ils sont échangés entre composants distants, il s'agit alors de spécifier une *configuration de messagerie qui soit faisable* sous l'ordonnancement de trames induit par le protocole MAC.

Plusieurs problèmes s'ajoutent à celui-ci :

- la détection, le diagnostic et lorsque c'est possible, la tolérance aux erreurs de transmission ; il s'agit d'une part de spécifier des modèles d'erreurs, d'établir des modèles de relations formelles entre fautes au niveau de la communication et défaillances au niveau du système, et de définir des mécanismes permettant de conserver autant que se peut le système dans un état sauf ;

- la prise en compte des politiques de gestion des communications par les contrôleurs du commerce (politique de gestion des files, en particulier) et l'intégration de mécanismes adaptatifs permettant d'assurer les propriétés en ligne à tout instant ;

Remarques : les composants applicatifs relèvent pour la majorité d'exécution périodique ou sporadique dont un pire cas peut se ramener à une exécution périodique ; la génération et la configuration d'ordonnancement faisable de tâches localement à chaque ECU et de trames échangées entre ECU relèvera donc du cadre périodique. Néanmoins, certains événements se produisent de façon apériodique et aléatoire ; ils peuvent être également caractérisés par des propriétés temporelles critiques et ils doivent donc être considérés dans les travaux.

Enfin, les deux paradigmes des réseaux de communication embarqués dans l'automobile doivent être pris en compte, il s'agit :

- des réseaux à priorité, comme CAN (Controller Area Network [ISO 11898] : dans ce cas, il s'agit de concevoir un frame-packing qui donne une messagerie faisable où chaque trame est décrite par les signaux ou fragments de signaux qu'elle contient, son identifiant qui fixe sa priorité lors de l'accès au médium de communication et sa règle d'émission (sa périodicité d'émission, par exemple) ;

- des réseaux guidés par le temps, dont l'accès au médium relève de la politique générale TDMA (Time Division Multiplexed Access) et pour lesquels, il s'agit de spécifier un frame-packing, une configuration de cycles TDMA ainsi que l'allocation des slots aux trames.

1.2 Caractérisation de la robustesse des services de communication d'un intergiciel

La sûreté d'un système peut s'exprimer sous forme de contraintes requises sur des paramètres des services de communication, en particulier, on peut citer :

- la régularité entre les fins de transmission de trames contenant des signaux émis périodiquement (gigue sur la fin de transmission des trames) ; cette propriété a un impact dans le cas des systèmes contrôlés en réseaux au sein du système électronique embarqué,

- la latence sur la transmission d'un signal, donc de la trame qui le contient,

- la capacité à détecter et/ou tolérer des fautes de transmission.

Les propriétés exigées sur les deux premiers paramètres peuvent s'exprimer sous la forme d'une borne à respecter (borne maximale sur la latence entre l'émission et la réception d'une trame ou borne maximale sur la gigue) ; les garanties demandées peuvent être strictes (respect des bornes par toutes les instances), fermes (respect des bornes au pire par un profil d'instances) ou souples (respect des bornes en moyenne sur les instances).

1.2.1 Robustesse aux contraintes de la plate-forme d'exécution

Les politiques d'ordonnancement locales et le protocole d'accès au médium de communication ont un impact important, même en absence d'erreurs de transmission, sur le respect des propriétés énoncées ci-dessus. Il est donc important de se donner les moyens de les vérifier. Par exemple, K. Tindell [Tindell *et al.* 1994] a proposé, dans le cas de réseaux à priorité comme CAN, une formule permettant de trouver le pire temps de réponse pour un ensemble de trames périodiques ou sporadiques. Cette formule a été remise en cause puis corrigée dans [Davis *et al.* 2007].

La pire latence d'un signal transmis dans une trame sur un réseau TDMA dépend uniquement de la longueur du cycle. [Wilwert 2005] propose, dans le cadre d'un système système de contrôle X-by-Wire où la référence conducteur est transmise de manière redondée via un réseau TDMA , de mesurer cette valeur grâce à un modèle Matlab/Simulink de la loi de commande connecté à un modèle du véhicule qui fournit un indicateur résultant d'une fonction des valeurs du roulis, tangage, lacet et écart à la trajectoire observées sur le modèle du véhicule. La longueur de cycle est introduite dans le modèle de la loi de commande. Il évalue alors les longueurs de cycles pour lesquelles la valeur de l'indicateur est admissible, signifiant alors que le système est contrôlable.

1.2.2 Robustesse aux fautes externes transitoires

Même dans le cas où la faisabilité d'une messagerie est prouvée, le traitement des erreurs au niveau du protocole peut avoir un impact sur les performances et donc sur la robustesse de l'application. Nous développons ci-dessous quelques études sur des modèles de fautes ainsi que des métriques associées dans le contexte des systèmes de communication considérés dans l'automobile.

Dans ce contexte, les premiers à avoir pris en compte les erreurs de communication étaient Tindell et Burns dans [Tindell *et al. 1994*] pour le protocole de communication CAN. Le modèle d'erreurs proposé correspondait alors à une seule rafale d'erreurs et un nombre borné d'erreurs isolées (séparées par une durée minimale paramétrable) pendant toute la durée de l'étude. On peut alors borner le temps de réponse maximal pour un nombre d'erreurs donné. Les formules sont directement dérivées des formules précédentes et du modèle d'erreurs proposé.

N. Navet a fait l'hypothèse dans [Navet *et al.* 2000] qu'il est irréaliste de donner une borne pour le nombre maximal d'erreurs pouvant survenir pendant un intervalle de temps. Il a ainsi proposé un modèle probabiliste d'erreurs dans le but d'obtenir la probabilité qu'une trame ne respecte pas

son échéance. Ce modèle permet de prendre en compte la fréquence d'arrivée des erreurs et leurs durées (rafales). La fréquence d'apparition est telle que le nombre d'occurrences suit une loi de Poisson, et la durée de la rafale peut entraîner une seule erreur ou une rafale. D'un point de vue mathématique, ce modèle est décrit par un processus de Poisson généralisé. Il présente l'avantage de prendre en compte les rafales d'erreurs sans bornes prédéfinies quant à leurs durées et associé à une technique statistique d'identification de paramètres à partir d'une campagne de mesure. Une métrique est également proposée : la probabilité de défaillance pire cas ou *Worst Case Failure Probability* ; les auteurs montrent comment évaluer cette probabilité.

En 2002, dans [Broster *et al.* 2002], puis dans sa thèse , I. Broster a remis en cause le modèle précédent, en raison de son pessimisme, et plus particulièrement le fait que Navet fasse l'hypothèse que, si une erreur se produit pendant le temps de réponse maximal d'un message, ce message est perdu. Or, ce temps n'étant qu'une borne sur le temps de réponse, il se peut que le message soit déjà arrivé avant l'occurrence de l'erreur. La deuxième source de pessimisme repose sur l'hypothèse selon laquelle une rafale d'erreurs peut affecter plusieurs messages consécutifs. Il propose alors d'utiliser un modèle "simplifié" qui ne prend pas en compte les rafales. Le dernier modèle d'erreurs proposé par Broster dans [Broster *et al.* 2004] nous donne ainsi le nombre d'erreurs k se produisant pendant une durée t. Ce dernier suivant une loi de Poisson. Il part du principe que la probabilité pour un message m de taille T_m (temps bit) d'être perdu est la probabilité qu'il soit affecté par au moins une faute pendant T_m. Il fait alors l'hypothèse que les fautes sont indépendantes, et que seule la perte des n copies du message entraîne une défaillance. Sur cette base, il propose le calcul de probabilité de défaillance.

Enfin, dans le cas de réseaux de type TDMA, [Wilwert 2005] propose d'identifier la tolérance maximale en termes de cycles corrompus pour un système de contrôle X-by-Wire où la référence conducteur est transmise de manière redondée via un réseau TDMA. Cette identification est faite à l'aide du modèle cité précédemment La prise en compte de la longueur du cycle TDMA et l'injection de fautes sur ces cycles sont appliqués sur le modèle de la loi de commande. L'observation sur une campagne de mesure importante a montré qu'il était possible de garder le véhicule dans un état sauf si moins d'un nombre k de cycles étaient consécutivement corrompus. Ce type de systèmes connus sous le nom de systèmes « consecutive-k-out-of-n:F ». Une forme récurrente a été fournie dans [Simonot-Lion *et al.* 2005, Simonot *et al.* 2005] pour évaluer la probabilité de défaillance, c'est-à-dire la probabilité d'avoir plus de k cycles consécutifs corrompus sous un profil d'erreurs de durée et de probabilité d'occurrences données.

1.3 Amélioration de la robustesse des services de communication d'un intergiciel

1.3.1 Prise en compte des contraintes de la plate-forme

Ainsi que nous l'avons vu, le problème peut se décomposer en 2 sous-problèmes : la configuration d'une messagerie, d'une part et la configuration locale des services de l'intergiciel gérant en ligne cette messagerie.

1.3.1.1 Configuration de messageries correctes et « optimales »

Dans cette section nous allons présenter quelques algorithmes de *frame packing réalisés hors ligne*. L'objectif de ce type d'algorithme est de construire chaque trame (à partir d'un ensemble de signaux ou fragments de signaux), et pour chacune d'elles de calculer leurs paramètres. Ceux-ci

sont : la taille (somme des tailles des signaux composant la trame), la période de transmission, l'échéance relative, et la priorité (si on utilise un réseau à priorités). Le but de tous les algorithmes présentés ci-dessous est de minimiser la consommation de bande passante. Toutefois, la faisabilité des trames n'est pas traitée par tous.

Réseaux à priorité

Volcano [Rajnak *et al* 1998, Casparsson *et al.* 1999, Rajnak 2005] est un outil commercial de construction de messagerie optimisée faisable. Il comprend deux volets : un algorithme de frame-packing exécuté hors ligne et un « intergiciel » qui applique les règles définies en ligne. Néanmoins, cet algorithme n'est pas connu à cause du caractère commercial de Volcano. Les algorithmes présentés ci-dessous sont le fruit de la recherche universitaire et sont donc publiques.

La première référence à un algorithme de frame packing, appelé *piggyback*, a été faite dans [Tindell *et al.*1994]. Le but des auteurs de ce travail était celui de minimiser la consommation de bande passante d'un ensemble de trames émises sur le réseau CAN, ainsi que le respect de la faisabilité. La stratégie utilisée était d'insérer des signaux ayant la même période de production dans une trame (celle-ci aurait la même période que les signaux). En plus, les auteurs ont prévu le cas où il y ait des signaux dont la production ne soit pas périodique. Dans cette situation, ils ont proposé l'utilisation d'une trame serveur périodique qui, au moment d'être construite, prendrait les signaux dont une occurrence ait été enregistrée.

Dans [Norström *et al.* 2000], deux heuristiques pour la construction et configuration de trames qui minimisent la consommation de bande passante sur les réseaux CAN et LIN ont été introduites. La première heuristique insère des signaux dans une trame jusqu'à ce que celle-ci ne puisse plus accepter de signaux. À ce moment là, une nouvelle trame est créée. La deuxième heuristique considère que les trames peuvent avoir différentes tailles. L'algorithme démarre avec une trame ayant la plus petite taille, et y insère des signaux jusqu'à que la trame n'ait plus de place libre pour recevoir un signal donné. À ce stade, un choix est fait : l'augmentation de consommation de bande passante causée par l'accroissement de la trame de façon à pouvoir accepter le nouveau signal, est comparée à celle causée par la création d'une nouvelle trame ayant la plus petite taille capable d'accepter le nouveau signal. L'alternative qui augmente le moins la consommation de bande passante est choisie. A partir de la création d'une deuxième trame, chaque signal à insérer est essayé dans toutes les trames existantes qui ont encore de la place libre.

Le travail présenté dans [Pop *et al.* 2005] a aussi traité le problème du frame packing. En particulier, ces chercheurs ont travaillé sur la construction et la configuration faisable de trames dans un système distribué, composé d'un réseau du type time-triggered, d'un autre du type event-triggered, et d'une passerelle qui lie les deux réseaux. La différence dans ce travail par rapport à ceux présentés précédemment est dans le modèle de l'application. En effet, le modèle de l'application utilisé est basé sur les graphes acycliques de processus. Une application est représentée par un ensemble de graphes de processus, où chaque processus (ou nœud) est une séquence de calculs qui démarre quand toutes ses valeurs d'entrée sont disponibles. Les arcs dans un graphe représentent les dépendances entre les processus qu'ils inter-connectent, et prennent la forme de signaux si les processus inter-connectés sont distribués. De plus, tous les processus et signaux présents dans un graphe ont la même période d'activation ou production. Finalement, les échéances sont attribuées au graphe, et non à chaque processus.

Deux algorithmes de frame packing ont été proposés dans [Santos-Marques *et al.* 2003] et leurs performances ont été évalués (en termes de minimisation de la bande passante et d'impact des

contraintes de fraîcheur sur la capacité à trouver des solutions). Les algorithmes proposés sont décrits rapidement ci-dessous :

- L'algorithme *Bandwidth-Best-Fit decreasing* (BBFd) repose sur les approches de résolution hors-ligne des problèmes de *bin packing*. Dans ce genre de problèmes, l'objectif est de minimiser le nombre de boîtes, étant donné un ensemble d'objets de tailles différentes qui doivent être placés dans ces boîtes. Dans notre contexte, le but est de minimiser la consommation de bande passante sans tenir compte du nombre de trames. L'idée clé de cet algorithme est, à chaque étape, de placer un signal dans la trame qui minimise au plus la consommation de bande passante supplémentaire causée par le signal. L'algorithme construit une seule solution dont les trames ont des caractéristiques, comme le pire temps de transmission, qui assurent le respect des contraintes de fraîcheurs des signaux transportés. Les priorités des trames sont allouées à l'aide de l'algorithme d'Audsley [Audsley 1991] dont l'applicabilité aux cas non-préemptif, c'est-à-dire pour des trames, a été montré. Dans le cas d'échec au niveau de l'algorithme d'Audsley, l'algorithme BBFd exécute quelques transformations sur les trames de façon à isoler les signaux les plus exigeants (ceux qui ont les contraintes de fraîcheurs les plus strictes).

- L'algorithme Semi-Exhaustive (SE) effectue un parcours exhaustif de l'espace des solutions. Il commence par construire sur chaque ECU la liste complète de tous les ensembles de trames possibles. Au contraire de l'algorithme BBFd qui peut être utilisé sur tous les problèmes, l'algorithme SE est seulement applicable sur des problèmes de moindre taille (moins de 12 signaux par ECU). Ensuite, l'algorithme trie la liste sur chaque ECU dans l'ordre croissant de la consommation de bande passante. Puis, il trie les ECU dans l'ordre croissant du premier ensemble de trames, celui qui minimise le plus la consommation de bande passante sur chaque ECU. Finalement, l'algorithme SE construit à la volée toutes les solutions possibles avec un ensemble de trames de chaque ECU, et arrête dès qu'une solution est faisable (algorithme d'Audsley).

Notons que les techniques précédentes concernent des algorithmes de configuration sur CAN hors-ligne. Zuberi et Shin [Zuberi et Shin 1995] proposent des travaux pour configurer dynamiquement des messageries sur CAN en faisant un compromis entre NP-DM et NP-EDF. L'échéance d'une trame est codée dans une partie de l'identifiant (l'autre est réservée pour garantir l'unicité des identifiants imposée par CAN). Le temps est divisé en époques et un processus périodique met à jour sur chaque station les échéances des trames relativement au début de cette époque. Meschi et al.[Meschi et al. 1996] montrent qu'il est possible d'utiliser NP-EDF. Ils proposent une implémentation de NP-EDF sur CAN, qu'ils ont améliorée dans [Di Natale 2000, Di Natale et Maschi 2001] L'idée de base, déjà suggérée par Zuberi et Shin, est de coder l'échéance de la trame en utilisant une grande partie de l'identifiant. Comme il est imposé que toutes les trames aient un identifiant distinct sur CAN, quelques bits sont réservés pour assurer l'unicité des identifiants. Une solution est d'allouer une valeur distincte à ces bits réservés pour chaque flux. Les auteurs proposent également des solutions pour limiter les erreurs de codage (inversion de priorité) et fournissent une analyse d'ordonnançabilité.

Une approche proposée par N. Navet est le lissage de flux sous contrainte temps réel [Navet 1999]. Celle-ci se place dans le cadre suivant : deux types de trafic coexistent, l'un à contraintes strictes dit temps réel, et l'autre à contraintes souples (non temps réel). L'objectif est alors de minimiser le temps de réponse du trafic non temps réel en garantissant le respect des contraintes du trafic temps réel. Dans cette optique, une politique d'ordonnancement à lissage de flux (« traffic shaping ») a été développée. Sa faible complexité permet son utilisation « en ligne ». De plus, elle

est compatible avec les contrôleurs de communication existants. L'idée de base est qu'il est possible de réduire le temps de réponse du trafic à contraintes souples si les périodes d'activité engendrées par le trafic à contraintes strictes sont « régulièrement » réparties dans le temps, créant ainsi des intervalles de temps durant lesquels la ressource (le processeur ou le medium de communication) peut être utilisé par du trafic non temps réel avec un minimum de délai. La politique proposée s'applique aussi bien à l'ordonnancement de tâches que de messages. Si le système est faisable avec BS (Background Scheduling), celui-ci le sera également sous cette politique.

L'utilisation d'offsets dans l'ordonnancement de messages, qui provoque une désynchronisation des flux de messages, permet d'obtenir de meilleures performances [Grenier *et al.* 2006]. Des gains en termes de temps de réponse sont réalisés car, de cette façon, le trafic est réparti plus uniformément dans le temps.

Réseaux de type TDMA

La configuration de messagerie et de cycle est réalisée, pour TTP/C par des outils propriétaires de la société TTTech. Aussi, il n'y a pas, à notre connaissance, de travaux publiés sur ces algorithmes.

FlexRay est un protocole hybride de type TDMA spécifiquement conçu pour l'automobile. Il concilie à la fois les communications de type *Time-Triggered* (segment statique) et *Event-Triggered* (segment dynamique) et propose des garanties déterministes pour l'ensemble des messages [Pop *et al.* 2007]. Il est très probablement le futur standard *de facto* des réseaux embarqués dans les véhicules. Dans [Pop *et al.* 2007], les auteurs proposent un modèle de l'application qui identifie les messages à placer dans le segment statique (ST) et les messages à placer dans le segment dynamique (DYN) ainsi que les tâches cycliques ordonnancées statiquement (SCS) et celles ordonnancées selon une politique à priorité (FCS). Le problème est alors, pour les messages ST et les tâches SCS de construire les tables d'ordonnancement et, pour les tâches ST, d'évaluer la faisabilité par évaluation du pire temps de réponse. Des algorithmes sont proposés pour les deux sous problèmes.

1.3.1.2 Déploiement des services de messagerie d'un intergiciel embarqué

La modularité et la portabilité des composants applicatifs sont des aspects importants dans le développement d'applications embarquées ; il est alors indispensable de découpler l'application des activités de gestion de la messagerie, comme par exemple, la mise en trame et l'émission des signaux à émettre. C'est le rôle d'un intergiciel. Cependant, un intergiciel générique de type CORBA, ne permet pas de maîtriser les services de communication qui préservent la conformité au modèle de frame-packing calculé hors ligne, et donc la faisabilité de la messagerie. Il faut donc générer un intergiciel ad hoc. La plupart des travaux relevant de ce problème propose un comportement de l'intergiciel asynchrone par rapport à celui de l'application. Entre autres, on peut signaler les travaux suivants.

ACE [Schmidt 1994, Schmidt 1997] (Adaptative Communication Environment) est un cadre de travail qui implémente des patterns pour la concurrence et pour la distribution. Cet intergiciel fournit un ensemble de wrappers (types primitifs de données qui offrent des méthodes permettant leur manipulation) et de composants en C++, qui sont ciblés vers le développement d'applications et de services de haute-performance et de temps-réel, en recouvrant un vaste éventail de plates-formes. TAO (The Ace Orb [Schmidt *et al.* 1998]) est un ORB ciblé pour des applications ayant

des besoins de qualité de service temps-réel (traitement de missions aéronautiques, applications multimédia, et simulations interactives distribuées). Cet intergiciel objet présente une conception extensible car son développement est guidé par un langage de patterns. Cette caractéristique permet à TAO d'être dynamiquement configuré de façon à respecter, d'une part, les besoins de qualité de service des applications, et, d'autre part, les caractéristiques de la plate-forme de communication utilisée. TAO présente une caractéristique essentielle au niveau de son développement : les patterns, ce que lui permet d'être extensible et modulaire. De plus, il semble contenir des services qui lui permettent de garantir des délais déterministes aux applications. Cependant, aucune indication relativement à son implémentation (le nombre de tâches essentiellement) n'est fournie ce qu'empêche toute analyse d'ordonnançabilité. Enfin, les services de communication en utilisant des mécanismes de nommage, et la configuration dynamique de ces services font de lui un candidat peu adapté à être un intergiciel automobile.

Les travaux présentés dans cet état de l'art suivent les propositions du consortium AUTOSAR et proposent une méthode de déploiement des services de messagerie. Deux aspects importants pour le développement d'un intergiciel doivent être abordés :

- l'aspect implémentation, qui sera traité en définissant une architecture logicielle optimale capable de grouper les spécifications de code de l'intergiciel, ainsi que son déploiement, et

- l'aspect configuration, qui se chargera de déterminer les paramètres qui permettront à l'image de l'intergiciel déployé, de s'exécuter tout en respectant les contraintes temporelles imposées sur les tâches et les signaux.

Les objectifs de l'intergiciel Volcano [Rajnak *et al.* 1998, Casparsson *et al.* 1999, Rajnak 2005] sont de fournir une assistance à la conception des systèmes de multiplexage véhicule (pour l'instant, autour des réseaux CAN [ISO 1198 1994] et LIN [LIN 2003]), indépendamment du protocole de communication sous-jacent. Volcano est caractérisé, ainsi que nous l'avons vu ci-dessus, par la garantie *a priori* des propriétés temporelles de la messagerie par la méthode de conception du système (garantie de respect des échéances sur les signaux), la flexibilité de la méthodologie donnant au constructeur la possibilité de mise à jour et re-configuration (upgrade) de la messagerie en phase de pré-production, et même après la phase de mise en marché du véhicule, et l'utilisation efficace des ressources. Un outil logiciel permet de générer le code Volcano optimisé pour un ECU cible. Cet outil fournit une API basée sur les signaux, supporte plusieurs protocoles réseaux, et permet la reconfiguration de la messagerie même après la compilation de l'application. Le modèle de communication implanté repose sur le modèle producteur/consommateur (ou éditeur/souscripteur pour Volcano). L'API cache tous les détails concernant le comportement du réseau, et ne doit permettre que la manipulation des signaux produits et consommés. Le problème principal de Volcano est qu'il présente une API qui ne cache pas à l'application l'existence d'un réseau de communication.

La méthodologie proposée dans [Santos-Marques *et al.* 2005] considère ces services d'émission et réception de trames et de signaux. D'autres services pourraient être offerts par l'intergiciel, comme la surveillance de la qualité de service par exemple. L'objectif sous-jacent final de la méthodologie est le développement d'un système qui respecte toutes les propriétés qui lui sont imposées, en particulier les propriétés de performances. Ceci passe par une caractérisation (activation, priorité) des tâches applicatives, des tâches de l'intergiciel et des trames échangées entre les ECUs. Dans ce problème, les tâches applicatives sont données (charge CPU, règles d'activation) mais leur priorité n'est pas définie. Pour leur attribuer des priorités, l'auteur propose au préalable de définir les caractéristiques des tâches qui exécutent l'intergiciel sur chaque ECU. Ces caractéristi-

ques permettent ainsi une quantification exacte de l'interférence provoquée par l'intergiciel sur les tâches applicatives. Cependant, les caractéristiques des tâches de l'intergiciel dépendent du résultat de la configuration de trames. L'algorithme qui détermine cette configuration repose sur un calcul du pire temps de réponse pour toutes les tâches (intervalle de temps maximal entre l'activation d'une tâche et sa fin d'exécution), c'est-à-dire sur un modèle de tâche applicative qui exhibe, déjà, les priorités de celles-ci. Ce raisonnement fait apparaître le fait que chaque activité élémentaire (caractérisation de tâches applicatives, caractérisation de tâches de l'intergiciel, caractérisation de trames) dépend du résultat des autres activités. Pour surmonter cette boucle d'interdépendance, la méthode définit un processus de configuration,ainsi :

- La première étape est l'activité qui configure les trames selon une des méthodes présentées en section précédente. Ses données d'entrées sont les caractéristiques des signaux (taille, contraintes de fraîcheur) et certaines caractéristiques des tâches applicatives (règles d'activation et charge CPU).

- La deuxième étape consiste en la caractérisation complète, dans un premier temps, des tâches de l'intergiciel puis, des tâches applicatives. La caractérisation des tâches de l'intergiciel est également conditionnée par le modèle générique d'implémentation de celui-ci ; ses données d'entrée sont l'ensemble des trames configurées et faisables ainsi que les composants logiciels à déployer dans ces tâches (méthodes, séquences de code). Le résultat est l'empreinte de l'intergiciel à savoir, un ensemble partiellement configuré de tâches (règles d'activation) ainsi que le code exécuté par ces tâches. Dans un deuxième temps, sont déterminées les priorités des tâches de l'intergiciel et des tâches applicatives afin d'obtenir un ordonnancement de tâches faisable sur chaque ECU.

Le développement d'un intergiciel pour une application et une distribution données est conditionné, ainsi que nous l'avons vu ci-dessus, sur le modèle générique d'architecture de l'intergiciel (modèle d'implémentation qui utilise, comme donnée d'entrée, le modèle des composants logiciels de cet intergiciel tels que définis, par exemple, par un diagramme de classes UML).

La méthodologie sur laquelle repose l'outil SYNDEX [Sorel 2004] vise le prototypage rapide et l'implantation optimisés d'applications distribuées temps réel embarquées. Elle est fondée sur des modèles de graphes, autant pour spécifier les algorithmes applicatifs et les architectures matérielles distribuées comportant un certain niveau de redondance, que pour déduire les implantations possibles en termes de transformations de graphes. Il s'agit de résoudre un problème d'optimisation consistant à choisir une implantation dont les performances, déduites des caractéristiques des composants matériels, respectent les contraintes temps réel. Dans le cas du temps réel critique les approches « hors ligne » sont privilégiées, et quand des approches « en ligne » sont utilisées, le nombre de décisions prises lors de l'exécution est minimisé, c'est-à-dire uniquement quand elles sont inévitables. Il est alors possible de générer automatiquement une part des exécutifs distribués temps réel à faible surcoût pour les composants processeurs. Afin de réduire la rupture entre la phase de spécification/simulation des automaticiens et la phase d'implantation en temps réel des informaticiens, et afin de minimiser la durée du cycle de développement des applications distribuées temps réel embarquées, les liens entre des langages haut niveau orientés métier et l'outil SynDEx sont étudiés. Notons que dans le cas de la distribution, les trames sont traitées comme des tâches.

1.3.2 Robustesse aux fautes externes

L'état de l'art pertinent pour SCARLET dans ce cadre concerne principalement les constructions de messageries, sous des hypothèses de fautes.

Dans le cas de bus à priorité, la politique DP (Dual Priority) permet de respecter les contraintes des messages temps réel en optimisant le temps de réponse des messages non temps réel. La principale différence entre cette politique et une politique à priorité fixe est le changement dynamique de la priorité des messages au cours du temps. Les messages ayant des contraintes strictes sont promus à la priorité la plus forte le plus tard possible, de manière à respecter leur échéance. Ainsi, DP permet d'avoir de bonnes performances pour le trafic à contraintes souples, si l'on fait l'hypothèse d'un médium fiable. Dans le cas contraire, un médium non fiable expose fortement le trafic temps réel à des dépassements d'échéances. Dans ce cadre, N. Navet propose un mécanisme simple permettant de garantir une qualité de service exprimée comme une probabilité de respect des échéances [Navet *et al.* 2000]. Cette proposition dégrade faiblement les performances de DP et donne des garanties probabilistes sur le respect des échéances. La qualité de service pourra être éventuellement individualisée en fonction de la criticité de chacune des trames. Des mécanismes qui permettent de fixer les paramètres du modèle d'erreurs « en ligne », et dont l'utilisation s'impose lorsqu'il n'est pas possible de déterminer *a priori* le niveau de perturbation sur le bus ou lorsque celui-ci est soumis à de fortes variations [Navet et Song 1999a] [Navet et Song 1999b]. Cette politique est basée sur un principe simple et efficace. Néanmoins, le changement dynamique de priorité, indispensable à son implantation, n'est pas disponible sur les contrôleurs de communication existants. Le développement d'une couche logicielle pourrait être envisagé si le contrôleur propose un service d'annulation de requête de transfert de données, ce qui, à notre connaissance, n'est proposé par aucun protocole de niveau « liaison de données ».

La configuration optimale de la communication inter-calculateurs dans les applications X-by-Wire pour un réseau TTP/C et dans le cas général TDMA a été étudiée dans [Gaujal et Navet 2003, Gaujal et Navet 2005]. Dans ces etudes, il est montré comment utiliser le concept de Fault Tolerant Unit (FTU) de manière optimale pour rendre un réseau de type TDMA robuste aux erreurs provenant de perturbations en rafales. Deux objectifs sont envisagés : minimiser la probabilité de perdre tous les réplicas d'un signal donné et, dans ce cas, il s'agit de répartir les réplicas dans le temps (propriétés de convexité de la probabilité de perte), ou minimiser la probabilité de perdre au moins un réplica et, alors, il s'agit de grouper les réplicas ensemble (utilisation de techniques de calculs de bornes majorantes). Une adaptation au cas particulier de TTP/C est fournie.

Alain Girault dans [Girault *et al.* 2003, Girault *et al.* 2004a, Girault *et al.* 2004b] adresse le problème de l'ordonnancement statique temps réel tolérant aux fautes sur des architectures multiprocesseurs distribuées et minimisant la longueur totale de l'ordonnancement généré. Les travaux sont basés sur la réplication active des tâches, où chaque tâche est répliquée sur au moins *Npf + 1* processeurs afin de tolérer *Npf* pannes. Il propose également une technique de calcul de date d'exécution pour chaque opération ainsi que les chiens de garde qui sont utilisés pour détecter les défaillances. Les algorithmes sont intégrés pour partie dans l'outil SynDEx afin de générer des systèmes tolérants aux fautes.

1.4 Références

[Audsley 1991] N. Audsley. Optimal priority assignment and feasibility of static priority tasks with arbitrary start times. Technical Report YCS164, University of York, November 1991.

[Broster *et al.* 2002] I. Broster, A. Burns, and G. Rodriguez-Navas. Probabilistic analysis of CAN with faults. In Proceedings of the 23rd Real-time Systems Symposium, Austin,USA, Dec 2002. IEEE.

[Broster *et al.* 2004] I. Broster, A. Burns, and G. Rodríguez-Navas. Comparing real-time communication under electromagnetic interference. In Proceedings of the 16th Euromicro Conference on Real-Time Systems, pages 45–52, Catania, Italy, July 2004. IEEE Computer Society.

[Casparson *et al.* 1992] L. Casparsson, A. Rajnak, K. Tindell, and P. Malmberg. Volcano - a revolution in on-board communications. Volvo Technology report, Volvo, 1999.

[Davis *et al.* 2007] R. I. Davis, A. Burns, R. J. Bril, J. J. Lukkien, Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised, Real-Time Systems 35(3): 239-272 (2007)

[Di Natale 2000] M. Di Natale, Scheduling the {CAN} bus with Earliest Deadline techniques, Proc. of the 21st IEEE Real-time Systems Symposium (RTSS 2000), 2000, Florida, USA.

[Di Natale et Maschi 2001] M. Di Natale and A. Meschi, Scheduling Messages with Earliest Deadline Techniques, Journal of Real-Time Systems, Vol 20, n°3, 2001.

[Gaujal et Navet 2003] B. Gaujal, N. Navet, Optimal Replica Allocation for TTP/C Based Systems, Proc. of the 5th FeT IFAC Conference (FeT'2003), Fieldbus Technology, Aveiro (Portugal), 7-8 July 2003.

[Gaujal et Navet 2005] B. Gaujal, N. Navet, Maximizing the Robustness of TDMA Networks with Applications to TTP/C, Real-Time Systems, Kluwer Academic Publishers, vol 31, n°1-3, pp5-31, December 2005.

[Girault *et al.* 2003] A. Girault, H. Kalla, M. Sighireanu, and Y. Sorel. An algorithm for automatically obtaining distributed and fault-tolerant static schedules. In *International Conference on Dependable Systems and Networks, DSN'03*, San-Francisco (CA), USA, June 2003.

[Girault *et al.* 2004a] A. Girault, H. Kalla, and Y. Sorel. A scheduling heuristics for distributed real-time embedded systems tolerant to processor and communication media failures. *International Journal of Production Research*, 42(14):2877-2898, July 2004.

[Girault *et al.* 2004b] A. Girault, H. Kalla, and Y. Sorel. An active replication scheme that tolerates failures in distributed embedded real-time systems. In *IFIP Working Conference on Distributed and Parallel Embedded Systems, DIPES'04*, Toulouse, France, August 2004. Kluwer Academic Publishers.

[Grenier *et al.* 2006] M. Grenier, J. Goossens, N. Navet, "Near-Optimal Fixed Priority Preemptive Scheduling of Offset Free Systems", Proc. of the 14th International Conference on Network and Systems (RTNS'2006), Poitiers, France, May 30-31, 2006.

[ISO 11898 1994] ISO. ISO 11898 - Road vehicles - Interchange of digital information - Controller Area Network for high-speed Communication. International Standard Organization, 1994. ISO 11898.

[LIN 2003] LIN Consortium. LIN (Local Interconnect Network) speci_cation package, version 2.0, September 2003. Available at http://www.lin-subbus.org.

[Meschi et al. 1996] A. Meschi, M. Di Ntale, and M. Spuri. Earliest Deadline message scheduling with limited priority inversion. Dans *les actes du 4ème International workshop on Parallel and Distributed Real-Time Systems (WPDRTS'96)*, 1996

[Navet 1999] N. Navet, "Evaluation de Performances Temporelles et Optimisation de l'Ordonnancement de Tâches et de Messages", INPL/LORIA,11/1999 (abstract in English - abstract in French).

[Navet et Song 1999a] N. Navet, Y-Q. Song, "Reliability Improvement of the Dual-Priority Protocol under Unreliable Transmission", Control Engineering Practice, Pergamon Press, vol. 7, n°8, 1999.

[Navet et Song 1999b] N. Navet, Y.-Q. Song, "Une Politique à Changement de priorité pour l'ordonnancement de messages dans des environnements bruités", Proc. of the 7ème Colloque Francophone sur l'Ingéniérie des Protocoles (CFIP'99), Nancy (France), 26-29 April 1999

[Navet *et al.* 2000] N. Navet, Y-Q. Song, F. Simonot, Worst-Case Deadline Failure Probability in Real-Time Applications Distributed over CAN (Controller Area Network), Journal of Systems Architecture, Elsevier Science, vol. 46, n°7, 2000.

[Norström *et al.* 2000] C. Norström, K. Sandström, and M. Ahlmark. Frame packing in real-time communication. Technical report, Mälardalen Real-Time Research Center, Sweden, 2000.

[Pop *et al.* 2005] P. Pop, P. Eles, and Z. Peng. Schedulability-driven frame packing for multicluster distributed embedded systems. Trans. on Embedded Computing Sys., 4(1):112_140, 2005.

[Pop *et al.* 2007] Traian Pop, Paul Pop, Petru Eles1, Zebo Peng1. Bus Access Optimisation for FlexRay-based Distributed Embedded Systems.

[Rajnak 2005] A. Rajnak. Volcano - enabling correctness by design. In Richard Zurawski, editor, Embedded Systems Handbook, pages 43.1_43.18. CRC Press Taylor and Francis Group, Boca Raton, FL, USA, 2005.

[Rajnak *et al.* 1998] A. Rajnak, K. Tindell, and L. Casparsson. Volcano communications concept. Technical report, Volcano Communications Technologies AB, 1998.

[Santos-Marques *et al.* 2003] R. Santos Marques, N. Navet, and F. Simonot-Lion. Frame packing under real-time constraints. In Proceedings of the 5th IFAC International Conference on Fieldbus Systems and their Applications (FeT'2003), pages 185_192, Aveiro, Portugal, July 2003.

[Santos-Marques *et al.* 2005] R. Santos Marques, F. Simonot-Lion, N. Navet, Development of an in-vehicle communication middleware, Object Oriented Modeling of Embedded Real-Time Systems, Post-proceedings of OMER 3, Heinz-Nixdorf Institute publisher, 2005.

[Santos-Marques *et al.* 2005] R. Santos Marques, N. Navet, and F. Simonot-Lion. Con_guration of in-vehicle embedded systems under real-time constraints. In Proceedings of the 10th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'2005), Catania, Italy, September 2005.

[Simonot *et al.* 2005] F.Simonot, F. Simonot-Lion, Y.-Q. Song, Dependability Evaluation of Real-Time Applications Distributed on TDMA-Based Networks, in 6th IFAC International Conference on Fieldbus Systems and their Applications - FeT'2005 (2005)

[Simonot-Lion *et al.* 2005] F. Simonot-Lion, F.Simonot, Y.-Q. Song, C. Wilwert, Quantitative Evaluation of the Safety of X-by-Wire Architecture subject to EMI Perturbations, in 10th

IEEE International Conference on Emerging Technologies and Factory Automation - ETFA'2005 1 (2005) 755-762

[Sorel 2004] Y. Sorel. Syndex: System-level cad software for optimizing distributed real-time embedded systems. *Journal ERCIM News*, 59:68-69, October 2004.

[Tindell *et al.* 1994] K. Tindell and A. Burns. Guaranteeing message latencies on controller area network (CAN). In CiA, editor, Proceedings of the 1st International CAN Conference (ICC'94), pages 2_11, Mainz, Germany, 1994.

[Wilwert 2005] C. Wilwert, Influence des fautes transitoires et des performances temps réel sur la sûreté des systèmes X-by-Wire, thèse de doctorat de l'INPL, Mars 2005.

[Zuberi et Shin 1995] K. Zuberi and K. Shin, Non-Preemptive Scheduling of Messages on Controller Area Network for Real-Time Control Applications, Proc. of the Real-Time Technology and Applications Symposium (RTAS'95), 1995, Chicago, IL, USA.

[Schmidt 1994] D. Schmidt. The ADAPTATIVE communication environment: an object-oriented network programming toolkit for developing communication software. In Proceedings of the 12th Annual Sun Users Group Conference, SUG, San Francisco, June 1994.

[Schmidt 1997] D. Schmidt. Applying design patterns and frameworks to develop object-oriented communication software. In Peter Salus, editor, The Handbook of Programming Languages. MacMillan Computer Publishing, 1997.

[Schmidt *et al.* 1998] D. Schmidt, D. Levine, and S. Mungee. The design of the TAO real-time Object Request Broker. Computer Communications, 21(4), April 1998.