In: Phœbus n ° 35 - 4 trimestre 2005, La revue de la sûreté de fonctionnement.
Ingéniere des connaissances et modélisation pour l'analyse de risques.
Tome 2 ISBN: 2-911221-38-9

GÉNÉRER DES SYSTÈMES DIAGNOSTIC AU LIEU DE LES PROGRAMMER

Oscar Dressler, Peter Struss

INTRODUCTION

La complexité des systèmes embarqués à bord des véhicules automobiles, ne cesse de croître, ainsi que le nombre de leurs interactions. Ce constat, combiné à l'explosion du nombre de variantes, favorise la mise en œuvre de compromis entre le niveau de qualité intrinsèque associé aux outils de diagnostics, et les efforts de développement consacrés aux systèmes proprement dits. Ce dont on a besoin, c'est d'une approche générale, exhaustive et systématique permettant la génération de ces systèmes de diagnostic, et de leur automatisation.

Cela doit être basé sur une représentation convenable de la connaissance sous-jacente aux différentes technologies mises en œuvre. Des bibliothèques contenant les modèles comportementaux des composants de ces véhicules doivent pouvoir piéger une part essentielle de cette connaissance. Des modèles de sous-systèmes automobiles peuvent être générés automatiquement à partir de tels éléments de librairie, et fournir la base d'une génération automatique, au lieu de conduire à programmer ces applications de diagnostic. La mise en œuvre de modèles explicites résulte en une amélioration et une assurance de meilleurs taux de couverture, et donc d'un meilleur niveau de sécurité. Nous expliquons comment mettre en œuvre ces solutions basées sur la modélisation, et démontrons leur applicabilité , en utilisant une étude de cas concernant un système embarqué à bord d'un véhicule Volkswagen Polo impliquant 4 unités de contrôle à travers un système CAN. Nous affirmons que la représentation de connaissance experte en termes de modèles le permet, indépendamment des facteurs temps, espace ou acteurs, ainsi que l'aide informatisée à la mise en œuvre de divers processus industriels tels que l'AMDEC, l'analyse de Testabilité et Diagnostabilité, la mise en place de sondes de test, la génération de test, ainsi que la mise au point d'ateliers de diagnostic.

CONTEXTE

Le processus de diagnostic de sous-systèmes automobiles devient de plus en plus difficile à assurer, et son informatisation renvoie à un défi technique encore supérieur, que ce soit sur un mode embarqué ou dans le cadre d'un atelier intégré. Cela déplace la difficulté au niveau de la production de logiciels de diagnostic , et pourrait apparaître comme un problème classique de qualité logicielle. Néanmoins, il apparaît que l'introduction d'applications logicielles de diagnostic sophistiquées embarquées à bord de véhicules automobile, peut être considérée comme une première étape, mais ne saurait s'imposer en tant que solution fondamentale aux problèmes de diagnostic rencontrés sur les systèmes actuels.

Relever ce défi technologique, et identifier des solutions requière une approche radicalement nouvelle basée sur la compréhension de la difficulté de ce processus. Les logiciels de diagnostic automobile ont pour mission de détecter, prédire, localiser, et identifier les dysfonctionnements sur les différents sous-systèmes et permettre leur réparation et leur remise en état. Cette affirmation évidente fournissent les clés pour déterminer la nature des défis à relever :

- les sous-systèmes automobiles se développent à travers une niveau de complexité toujours croissant, par ce que leurs composants sont de plus en plus sophistiqués, et parce qu'ils intègrent des logiciels,
- les sous-systèmes interagissent à la fois à travers leurs interactions physiques, mais aussi l'échanges d'informations transmises entre les applications logicielles embarquées, produisant des interactions de plus en plus complexes,
- en particulier, cela peut causer de multiples effets issus de défauts potentiellement distants des causes origines, et apparaissant du fait plus de ces multiples interactions plutôt que de la logique de décomposition hiérarchique entre composants,
- générer des logiciels de diagnostic susceptibles d'anticiper ces événements devient de plus en plus urgent, du fait des exigences croissantes sur le niveau de sécurité des véhicules, ainsi que leur impact sur l'environnement,
- par dessus tout, les différents types de sous-systèmes et de composants conduisent à de multiples déclinaisons, liées aux différents fournisseurs, conçues pour différentes technologies, renvoyant à de nombreuses fonctionnalités, typologies de véhicules, versions, etc..., ce qui nécessite de rendre compte de toutes ces variantes et déclinaisons dans le logiciel, et le processus de développement du logiciel.

Finalement, on est confronté à une tâche complexe, dont les coûts ne cessent de croître, du fait de ses aspects répétitifs. Puisque nous parlons de logiciels de diagnostic, le recours à des techniques d'ingénierie logicielle doit être d'un grand recours. Bien que cela soit indéniable, il faut aussi reconnaître que toute technologie logicielle réduite aux seuls aspects du processus de production logicielle et à la seule

valeur intrinsèque du logiciel produit, ne peut fournir de solution fondamentale : si cela n'inclut pas la prise en compte des sous-systèmes physiques eux-mêmes, cela ne peut pas refléter et gérer l'origine de la complexité et l'aspect répétitif de la tâche. Ce dont on a besoin, c'est d'une approche qui permet l'exploitation et le traitement effectif de la connaissance et de l'information concernant les technologies elles mêmes, les sous-systèmes physiques du véhicule, leurs composants, ainsi que leur structure. Nous devons baser le développement du logiciel de diagnostic sur des modèles informatiques des systèmes du véhicule.

Dans cet article, nous ne nous conterons pas seulement d'insister sur les idées clés et les fondements scientifiques des systèmes de diagnostic basés sur la modélisation (paragraphe 3), mais nous montrerons également la maturité de cette technologie dans les paragraphes suivants, en présentant les résultats d'une étude de cas, sur le diagnostic embarqué du système confort d'une Volkswagen Polo. Le diagnostic des quatre portes ainsi que leur système de communication à travers le réseau CAN a été généré automatiquement et complètement depuis un modèle aussi compacte qu'un code C élaboré pour un composant ECU (« Electronic Control Unit »). Finalement, nous montrons que, au delà des tâches de diagnostic, la « technologie » présentée dans cet article et basée sur une approche de modélisation, fournit les bases d'une intégration horizontale de différents processus élémentaires, se déroulant pendant la totalité du cycle de vie d'un véhicule, et commençant au début des étapes de conception.

GÉNÉRATION AUTOMATIQUE DE CODE POUR UN DIAGNOSTIC EMBARQUÉ

1999 : Etude de Faisabilité sur une Volvo 170

La faisabilité d'un système de diagnostic embarqué basé sur la modélisation fut démontrée la première fois dans le cadre du projet Brite-EuRam (Diagnostic de Véhicule Basé sur la Modélisation). Volvo Car Corporation, Robert-Bosch GmbH, OCC'M Software GmbH, ainsi que l'Université Technique de Münich développèrent un démonstrateur de détection et de diagnostic embarqué pour les symptômes liés à des dégagement de fumées noires, sur des moteurs turbo diesel (Volvo 850 TDI). Le véhicule de démonstration avait plusieurs cas de défaillance susceptibles d'être activés à travers une boîte à panne (fuites, défaillances mécaniques, pannes de senseurs). Les signaux des senseurs et de l'actuateur étaient transmis depuis une ligne série à un PC portable sur lequel tournait le système de diagnostic basé sur la modélisation (Figure 1). Les signaux traités étaient la pression atmosphérique, la pression de poussée, le flux d'air, la vitesse du moteur, le cycle de fonctionnement de la valve du moteur, la quantité d'essence injectée, c'est à dire les signaux habituels issus des ECU, sans qu'il ait fallu y ajouter des capteurs supplémentaires pour des fins spécifiques de diagnostic.

Figure 1 : PC portable supportant le logiciel de diagnostic basé sur la modélisation sur le véhicule de démonstration VOLVO.

L'application de diagnostic était basée sur des modèles simulant le comportement physique nominal des composants du turbo diesel et effectuait la localisation des défaillances avec une performance respectant les exigences de temps de réponse (cf Sachenbacher-Struss-Weber 00).

Ce prototype a démontré la faisabilité de principe de cette approche, mais, dans la mesure où ce développement s'est déroulé dans le cadre d'un projet de recherche et fonctionnait sur un PC Pentium sous Windows, deux questions ont été soulevées :

- Comment la production de telles solutions peut-elle être introduite dans le processus industriel d'un développement de logiciel embarqué ?
- Est-il possible de faire tourner des applications de diagnostic basées sur la modélisation sur des supports présentant les limites en mémoire et en vitesse de traitement d'ECUs standards ?

2004 : Génération Automatique de Code sur un ECU

Ces défis ont été remportés ces dernières années et les résultats ont été évalués sur une étude de cas, en utilisant un démonstrateur physique de « système confort » d'une Volkswagen polo qui avait été créé dans le cadre du projet STEP-X (Harms et al. 03).

Ce démonstrateur comportait un « rack » muni de quatre lève-vitres et deux miroirs associés à un système de positionnement, ainsi que les switchs permettant d'actionner ces éléments (Figure 2).

Les fonctions sont contrôlées par quatre ECUs (un par porte) qui sont connectés à travers un réseau CAN. Afin d'évaluer le diagnostic, on a pu pratiquement actionner une centaine de fautes, comprenant entre autres des court circuits et des circuits ouverts sur les moteurs électriques, des défaillances de senseurs concernant les vitres, des claquages de résistances de commutation, des défauts du réseau CAN, ainsi que des problèmes de connectique sur les différents ECUs.

Pour le benchmark, un cinquième ECU, basé sur un micro processeur « Infineon C 167 », et supportant l'application de diagnostic, a été connecté sur le réseau CAN. Des messages CAN intégrant entre autres une cinquantaine de signaux, ont été transférés au micro processeur de diagnostic toutes les 50 ms.

Figure 2 : démonstration d'un « système confort » dans le cadre du projet STEP-X

Le logiciel qui effectue le diagnostic basé sur ces messages est un code en C qui est automatiquement produit par un générateur de code basé sur les modèles des quatre sous-systèmes, de leurs systèmes de communication, ainsi que de leur alimentation électrique (Figure 3). Cela signifie que non seulement, le logiciel détecte et identifie les défaillances au niveau sous-système (lèves vitres et panneau de configuration respectifs), mais également au niveau des sous-systèmes en interaction et interdépendants.

Il faut noter que la décision d'implémenter le logiciel de diagnostic sur un ECU séparé, n'a pas été due à une question de principe, mais essentiellement à des raisons pragmatiques, en particulier, le fait qu'il était nécessaire de développer localement le logiciel de contrôle d'une part et de diagnostic d'autre part. Habituellement, les applications de diagnostic reliées aux sous-systèmes peuvent être lancées sur les ECUs respectifs, et le diagnostic centralisé sur un ECU séparé et rajouté aux ECU existants.

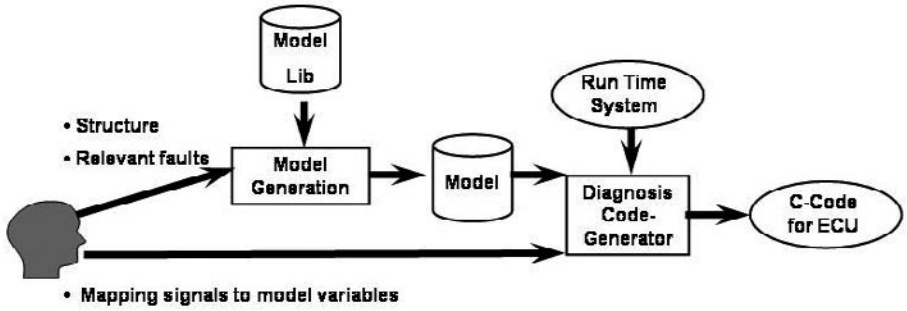

Figure 3 : Génération Automatique de systèmes de diagnostic embarqués

L'approche de génération de code a un impact majeur sur le processus de développement : toute modification ou évolution des systèmes physiques à diagnostiquer, consistant par exemple en un échange de composant ou un changement de structure, nécessite exclusivement les actualisations correspondantes du modèle ainsi que l'activation des procédures de génération automatique, afin que soient prises en compte ces modifications dans le processus de diagnostic.

Aucune modification directe du logiciel n'est requise, mais il est bien sûr possible de vérifier et d'interfacer le code avec d'autres logiciels de contrôle.

Le code C généré, une fois compilé, est particulièrement compacte, et ne nécessite que 25 kB de mémoire au total, c'est à dire en y incluant le pré traitement du signal, le modèle du système compressé, ainsi que les algorithmes de diagnostic. Même pour l'implémentation de la version centralisée du système de diagnostic la réalisation du système temps réel sur le processeur tournant à 19,5 MHz, n'a utilisé que 20% des capacités maximales en performance. Une caractéristique importante du système de diagnostic est qu'il n'est pas restreint aux pannes simples : même pour chaque sous-système élémentaire, il a pu traiter des cas de pannes multiples d'ordre arbitraire. Dans le contexte de ce démonstrateur, il a délivré l'information complète de diagnostic, en

termes de codes défauts (un tous les 50 ms), chacun de ces codes défauts représentant le jeu complet des défaillances possibles dans cette fenêtre de temps (et pas seulement un jeu de symptômes tels que des codes défauts courants).

Cela pourrait faire l'objet ultérieurement de différents traitements, filtrages, comptages statistiques...

Une évaluation des résultats de diagnostic basés sur le jeu complet des cas de pannes simulées a été réalisée en coopération avec l'Université Technique de Braunschweig ainsi que la société Carmeq GmbH. Un taux de réussite supérieur à 95% a été constaté, chaque cas de succès correspondant à une situation où le jeu de toutes les pannes possibles générées contenait la défaillance courante (à condition que la panne puisse être détectée pour des raisons physiques), et ne contenait pas des défauts incohérents avec les informations relevées.

Notre analyse des 5% cas restants de diagnostic incorrect révélèrent que qu'ils étaient dus à des limitations des modèles physiques (plutôt qu'à des limitations ou erreurs des algorithmes de diagnostic).

Le démonstrateur fut montré à la conférence Euroforum « Diagnostic de systèmes multiplexés automobiles » à Stuttgart, les 7 et 8 juillet 2004.

Ce benchmark a montré qu'une étape majeure a été franchie :

- les solutions basées sur la modélisation et concernant le diagnostic embarqué sont suffisamment compactes et performantes pour satisfaire des contraintes d'utilisation temps réel sur des ECU standards,
- elles fournissent des possibilités systématiques de génération automatique de code, et donc, une amélioration sensible du processus de développement d'applications de diagnostic sur des systèmes embarqués.

Dans les paragraphes suivants, nous allons aborder les principes sous tendant ce type de solutions, ainsi que leurs conséquences principales.

PRINCIPES DU DIAGNOSTIC BASÉ SUR LA MODÉLISATION

Pour comprendre pourquoi et comment les approches basées sur la modélisation destinées à la résolution de problèmes en général, et au diagnostic en particulier, représentent un véritable potentiel de révolution dans le domaine de la génération de logiciel, nous posons d'abord la question de savoir ce qui permet à un expert humain de réaliser un diagnostic, et de générer du code de diagnostic pour tous types de variantes associées à des systèmes connus, et même pour des nouveaux types de systèmes.

Manifestement, il s'agit de sa compréhension de comment ces systèmes se comportent et comment l'observation de ce comportement ou d'aspects mesurables renvoient à la présence de fautes possibles. Le fait qu'il soit capable de prendre en compte des innovations de conception, par exemple pour un circuit électrique, une nouvelle structure, montre qu'une partie de cette connaissance est générique. Cette partie

correspond essentiellement à la connaissance du comportement des différents éléments qui constituent le système. Cela s'applique de manière répétitive à chaque constituant élémentaire, à sa structure individuelle, et à ses interactions vis à vis des autres composants.

Aussi longtemps que cette connaissance fondamentale sera seulement disponible dans la tête des développeurs de logiciels de diagnostic, la production de ces applications logicielles de diagnostic restera une tâche répétitive, ce qui veut dire : coûteuse. Toute solution informatisée traitant de cet aspect, se doit de déplacer cette connaissance à travers une représentation explicite dans un programme informatique, et conduire ce programme à utiliser cette connaissance pour générer automatiquement les applicatifs de diagnostic, de manière à affranchir l'opérateur de cette tâche délicate et fastidieuse. C'est exactement ce que le diagnostic basé sur la modélisation effectue.

Le fait d'utiliser des modèles de systèmes pour générer des applications de diagnostic nécessite le recours à des algorithmes de diagnostic qui sont génériques, c'est à dire indépendants des systèmes à diagnostiquer. C'est la seconde contribution des diagnostics basés sur la modélisation.

En tant que résultat, l'élément de logiciel associé à la tâche de diagnostic (application de diagnostic) est séparé et indépendant de l'objet de cette tâche (le modèle du système qui doit être diagnostiqué), ce qui permet :

- la réutilisation d'algorithmes dédiés de diagnostic pour différents types de systèmes, et en particulier la génération automatique des applications de diagnostic,
- la réutilisation de modèles spécifiques de systèmes à travers différentes tâches (intégration horizontale).

Il est évident que cette séparation a un effet important sur l'utilité et la compétitivité économique de ce type de solution au sein des applications industrielles, et la distingue clairement d'autres approches qui exploitent aussi des modèles de systèmes, mais qui interagissent avec des aspects procéduraux entre éléments. Entre temps, nous allons analyser les algorithmes de diagnostic génériques et mettre en évidence quelques uns des principes de modélisation.

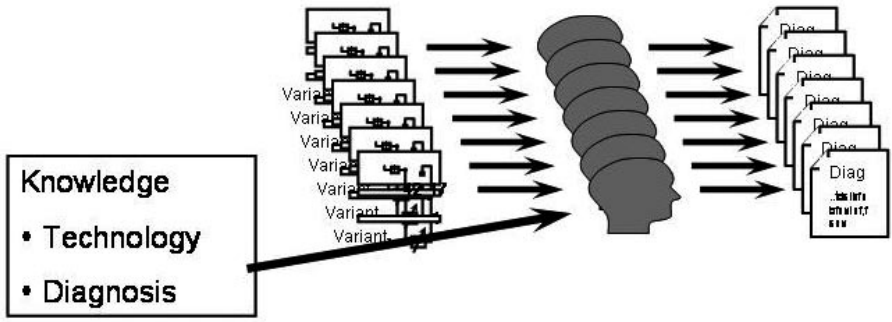

Figure 4 : Processus répétitif de la génération manuelle de systèmes de diagnostic

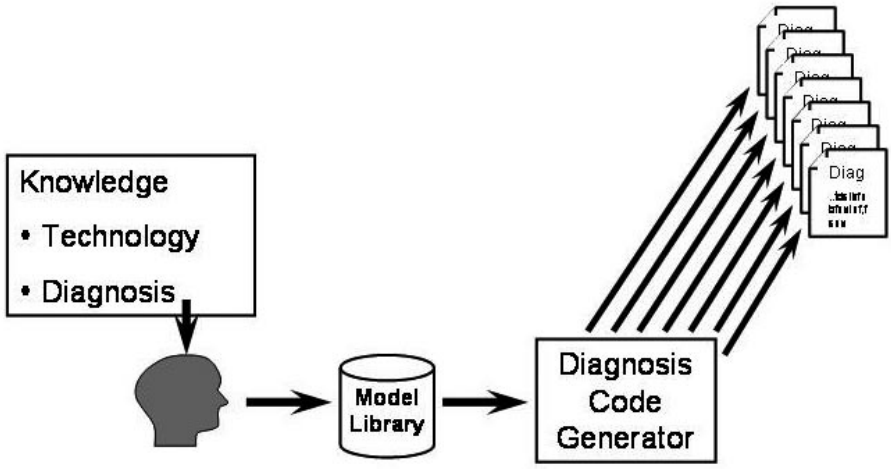

Figure 5 : Génération automatique de systèmes de diagnostic

Les fondements d'un diagnostic basé sur la modélisation

Nous expliquons la base des algorithmes de diagnostic génériques, en utilisant un exemple trivial comprenant une pompe commandée mécaniquement, et aspire de l'air depuis un container, et en supposant que la commande pilotant la conduite mécanique est connue. Si nous supposons un scénario où il y a une commande qui requièrent une accélération de la pompe, alors que les senseurs indiquent une augmentation de flux dans la pompe, ainsi qu'une chute de pression dans le container, nous sommes certains d'une présence de défaut, et en outre, nous pouvons supposer que la pompe ou le container présentent une fuite, ou bien encore que le capteur de pression est défaillant :

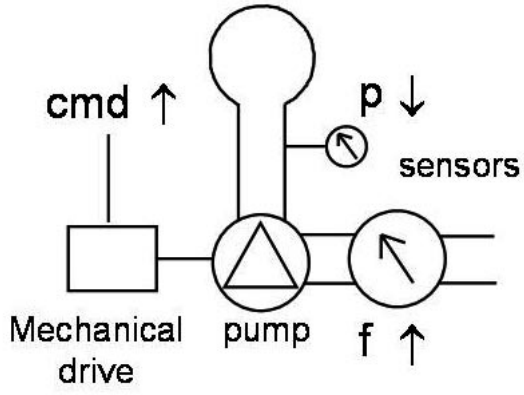

Figure 6 : Un « petit » problème de diagnostic

Comment un algorithme alimenté de modèles des comportements physiques des composants, avec en outre des informations concernant leurs connexions pourrait-il obtenir ce résultat ? Le cœur de cet algorithme vérifie la cohérence des éléments du modèle de système avec les informations observées (ce qui explique pourquoi cette approche est qualifiée de « basée sur la cohérence »). Par exemple, supposons que le comportement correct de la pompe, du container, et des deux capteurs se contredisent, parce que les modèles de comportement nominal des ces composants impliquent que les deux valeurs indiquent les mêmes tendances (signe de la dérivée première). Une implication logique de cette vérification est :

NOT OK(pompe) OR OT OK(container)
OR NOT OK (capteur pression) OU NOT OK (capteur de flux)

Qui indiquent la détection d'un défaut . De la même manière, les modèles de la commande mécanique, de la pompe, du container, et du capteur de pression sont tous ensemble en conflit avec la commande, et la mesure de pression, ce qui implique :

NOT OK(commande mécanique) OR OT OK(pompe)
OR NOT OK (container) OU NOT OK (capteur de pression)

Ensemble, ces deux conflits impliquent :

NOT OK(pompe) OR OT OK(container) OU NOT OK (capteur de pression)
OR NOT OK (capteur pression) AND NOT OK (commande mécanique)

Ce qui représente une localisation incomplète du défaut. Nous réalisons que ce résultat ne reproduit pas seulement les hypothèses du diagnostic présenté ci-dessus, mais illustre également que cet algorithme peut générer facilement des hypothèses de pannes multiples (en suspectant le capteur de flux et la commande mécanique de la pompe). En outre, nous insistons sur le fait que la localisation de pannes peut être basée sur des modèles qui représentent le comportement nominal exclusivement, et ne nécessite pas systématiquement les modèles des défauts composants. Ils sont uniquement requis si le défaut doit être identifié ultérieurement, et ils peuvent être vérifiés du point de vue de la cohérence à partir des mêmes observations. Par exemple, le modèle d'une pompe bloquée en position fermée sera réfuté, alors que celui d'une pompe présentant une fuite sera conservé. (Pour une présentation plus détaillée des fondements théoriques cf Dressler Struss 96)

Cet exemple trivial couramment cité est suffisamment éloquent pour illustrer la possibilité de générer des hypothèses de diagnostic basées un modèle de comportement, et de manière plus importante, pour suggérer que :

- cela peut être formalisé dans une théorie logique rigoureuse,
- cela peut être transformé en algorithmes indépendants du contenu spécifique de chaque modèle, et donc du système et du domaine auquel il fait référence.

Dans le projet STEP-X, les aspects électriques, mécaniques, et communication ont été traités de la même manière, par des algorithmes identiques. Et les mêmes principes ont été appliqués à une variété de systèmes techniques, allant des circuits numériques, aux systèmes d'alimentation électrique « haute tension » , ainsi qu'aux systèmes de stockage de fluide.

L'exemple montre également qu'un modèle (ainsi que son traitement) adaptés à un certain diagnostic, doivent avoir certaines propriétés que nous allons discuter dans ce qui suit.

Les langages de modélisation dits « compositionnels »

Les algorithmes évoqués dans le paragraphe précédent sont basés sur la reconnaissance de parties incohérentes identifiées dans le modèles global du système. Un modèle de type « boîte noire » de l'ensemble du système pourrait être suffisant pour la détection de fautes, mais serait de moindre utilité pour la localisation de défaut. Pour atteindre cet objectif, nous avons besoins de modèles dits « compositionnels », c'est à dire de modèles qui permettent la mise en œuvre d'algorithmes susceptibles d'identifier l'origine d'incohérences détectées, en termes d'entités du système responsables de ces incohérences.

Cette propriété vérifie une exigence importante qui apparaît dans le cadre du développement de logiciel : étant donné un algorithme de diagnostic générique, l'étape principale pour produire des diagnostiques efficaces sur chaque système spécifique, consiste à disposer des modèles spécifiques correspondants. Dans la mesure où il faut pouvoir traiter l'ensemble des variantes, mais aussi s'affranchir au maximum des tâches répétitives, il est hors de question de produire chaque modèle spécifique à partir de « zéro ». La solution consiste à capitaliser des modèles de (types) de composants au sein de bibliothèques, et faire en sorte que le modèle système spécifique soit généré automatiquement à partir de la « composition » de modèles élémentaires issus de cette bibliothèque. Ainsi, la production d'un système de diagnostic correspondant à un système spécifique relève de deux étapes :

- tout d'abord le modèle système est généré à partir de la donnée de sa structure de décomposition, et des modèles élémentaires issus de la bibliothèque,
- puis ce modèle est compilé pour donner lieu à la production d'algorithmes de diagnostic donnant lieu à la génération d'un code en C pour implémentation sur l'ECU.

Une fois que la bibliothèque existe et contient les modèles nécessaires, l'édition de la structure de décomposition du système, ainsi que son paramétrage constitue l'étape essentielle du développement de systèmes de diagnostic. Ce qu'il reste à faire, est de spécifier comment les signaux disponibles doivent être associés aux variables descriptives du modèle.

Ces caractéristiques :

- la composition automatique du modèle système depuis les objets d'une bibliothèque,

- la réutilisabilité de ces objets de bibliothèques pour différents types de systèmes,

sont essentiels pour obtenir des gains importants dans le processus de développement, et, une nouvelle fois, permettent aux différents solutions de diagnostic basées sur la modélisation de se distinguer les unes des autres.

Le taux de réutilisation des objets de bibliothèques est souvent influencé par le fait que, pour des raisons de diagnostic, souvent une description qualitative du comportement des composants suffit.

Modélisation Qualitative

L'exemple développé dans l'avant dernier paragraphe montre que souvent, une information infime comme le signe de la dérivée d'une quantité mesurée, suffit pour déduire au moins un premier jeu d'hypothèses de diagnostic. Utiliser des valeurs numériques n'améliorerait pas le résultat de diagnostic dans ce cas. Ainsi, le modèle comportemental doit seulement représenter des variations pertinentes par rapport à la discrimination correcte de modes de comportements « défaillants »

Pour donner un autre exemple issu du démonstrateur « STEP-X » : l'information issue de la position des fenêtres détectée par les capteurs à effets de Hall peut être discrétisée en les états « haut », « bas », et « moyen » , ce qui suffit complètement à vérifier leur cohérence, ainsi que leur sens d'évolution. Cela transforme le modèle en support de représentation générique et réutilisable, dès lors qu'on réussit à effectuer des correspondances avec ces trois états discrets. Cette mise en correspondance peut être également établie de manière dynamique avec des états évoluant en temps réel sur le système (même si cela n'a pas été implémenté sur le démonstrateur).

En outre, une réutilisabilité accrue des modèles qualitatifs est essentielle à l'efficacité et la compétitivité économique des approches de diagnostic basées sur la modélisation ; c'est la condition essentielle d'obtention :

- d'une représentation de modèle susceptible de générer des applications de diagnostic efficaces et cohérents,

- d'une représentation compacte de modèles,

- d'une réduction significative d'efforts informatiques et de modélisation, car les données d'entrées peuvent être approchées par des valeurs discrètes, et seulement des modifications de cette information qualitative nécessitent d'être actualisées dans les algorithmes de diagnostic (ce qui a été nécessaire pour l'exploitation du modèle temps réel concernant le moteur diesel VMBD).

Perspectives

Les travaux présentés ici montrent que cette technologie est à présent mature et permet de changer les pratiques dans le développement d'applications de diagnostic. Les techniques présentées permettent d'envisager la production automatique de modèles et d'applications de diagnostic embarquées, dans la mesure où des bibliothèques de modèles sont disponibles. Cela représente le prochain point de focalisation pour le déploiement réel de cette technologie dans l'industrie.

Ce travail n'est pas à sous-estimer : en effet, il permet de se demander si le bénéfice apporté par la génération de ces applications de diagnostic justifie un tel effort. Comme nous l'avons souligné, la réutilisabilité ne s'applique pas seulement aux algorithmes de diagnostic mais également aux modèles. Une bibliothèque de modèles peut-être considérée comme un référentiel de connaissances propriétaires sur un ensemble de technologies. Sous cette forme, la connaissance devient explicite, formalisée, et disponible vis à vis des experts humains, et des générateurs d'applications de diagnostic, indépendamment des facteurs temps, espace ou ressources humaines. Les modèles deviennent une base importante de systèmes de référence pour une intégration horizontale des processus d'analyse permettant l'échange de connaissances et le contrôle de sa cohérence.

Un modèle qui rassemble la connaissance sur le comportement nominal et dysfonctionnel de composants et systèmes, et qui permet la génération d'applications de diagnostic est aussi utile pour la réalisation d'autres tâches. Par exemple, l'AMDEC (Analyse de Modes de Défaillances de leurs Effets et de leur Criticité) peut également être automatisée en permettant la propagation de défaillances apparues sur chaque composant. AutoSteve, système développé et distribué par Mentor Graphics permet la génération automatique de ces AMDEC sur les systèmes électroniques, ainsi que l'Analyse de Circuits Insidieux (MentorGraphics 4). Par ailleurs, des sociétés de l'industrie aéronautique et spatiale, en collaboration avec des fournisseurs de logiciels et des universitaires ont développé un atelier générique pour la génération d'AMDEC dans les domaines hydrauliques, mécaniques et électriques, dans le cadre du projet AUTAS (Picardi et al. 04).

Encore une fois, la possibilité de formaliser des modèles à un niveau qualitatif sans description numérique détaillée, est essentielle pour effectuer ces analyses en phase amont du processus de conception, et pour générer des résultats sur des classes de défaillances et défauts.

Ainsi, la technologie est une contribution fondamentale, en particulier, du point de vue de la génération des systèmes de diagnostic en phase amont du cycle de conception des systèmes. Le projet Européen IDD (Integrated design Process for Onboard Diagnosis) a rassemblé différents acteurs du monde automobile, afin de produire une boîte à outils permettant de réaliser des analyses de diagnosticabilité et de testabilité basées sur la modélisation, mais également de générer des applications de diagnostic (y compris à base d'arbres de décision) à partir d'un même modèle (Dressler-Struss 03). Ce projet inclut une étude de cas qui met en évidence

l'intérêt d'approches basées sur la modélisation pour des systèmes présentant de multiples interactions. Les architectures multiplexées actuelles impliquant des équipements issus de multiples fournisseurs et présentant de nombreuses variantes en sont un exemple typique. Ces interactions multiples affectent les effets de défaillance, leur diagnosticabilité... La possibilité de formaliser et d'exploiter des modèles à un haut niveau d'abstraction constitue la base d'obtention d'un outil logiciel permettant à un utilisateur de faire face à ces problèmes combinatoires et de prendre des bonnes décisions.

Enfin, au delà de l'intégration d'activités reliées à la conception et au diagnostic embarqué, il est évident que les approches basées sur la modélisation facilitent les tâches de soutien dans le domaine de la production et du contrôle qualité (génération de tests), ainsi que de l'après vente (atelier de diagnostic).

En résumé, la technologie présentée dans ce papier ne remplace pas l'effort actuel consacré aux méthodes d'ingénierie logicielle avancée pour le développement d'applications de diagnostic embarquées. Bien plutôt, elle présente des solutions basées sur la formalisation systématique de connaissances destinée à maîtriser la complexité du processus de développement provenant du fait que le logiciel doit être développé en relation avec son environnement mécatronique, associé à toute sa complexité et ses multiples déclinaisons.

Oskar Dressler
Peter Struss
OCC'M