

ARCHITECTURE LOGICIELLE POUR UN SYSTEME DE DETECTION DES PANNES MOTEUR

Myriam NOUREDDINE

Institut d'Informatique, Université des Sciences et de la Technologie d'Oran
B.P 1505 Oran El-M'Naouer , Algérie

Abstract: We present the software architecture for a detection system of motor breakdowns in the mechanics domain. We describe the architecture entities of the system and the relationships between them into a formalism support. This formalism is integrated at the design step of lifecycle software.

Nous présentons l'architecture logicielle d'un système de détection des pannes moteur dans le domaine de la mécanique, à travers la description de ses entités et des liens entre ces entités.

Notre démarche, intégrée à l'étape de conception, se fait suivant deux étapes.

Premièrement, nous spécifions le système suivant un méta_modèle qui identifie dans l'ensemble mécanique les systèmes source de panne: nous nous préoccupons uniquement des pannes induites par l'environnement du moteur [5] par l'exploitation des symptômes constatés. Ces symptômes sont généraux [2,3], ce qui va permettre, ensuite, la génération des pannes induites par les pièces principales. La description statique de l'environnement est proposée ainsi que sa description dynamique vue à travers la stratégie de détection des pannes qui permet de produire en sortie la pièce responsable de la panne. Des exemples ponctuent cette première partie.

Nous appliquons ensuite notre méthodologie de construction [4] d'une architecture logicielle au système ainsi spécifié; celle-ci s'appuie sur le concept d'unité de description, noté UD, [4] qui résulte de l'unification entre la notion de modèle et les entités de conception. Nous proposons:

- la description du modèle, UD de top-niveau (niveau 1) ou UD0, qui représente le système global de détection des pannes moteur (définition de tous ses objets et de sa fonction).
- le processus de construction de la hiérarchie d'UD, à partir de l'UD0. La construction se fait suivant une approche descendante de décomposition des données, puis des opérations [1]: mise en place de UDi ($1 \leq i \leq 6$). Des relations entre les différentes UD peuvent alors se créer.
- la définition formelle et textuelle de chaque UD suivant un langage support [4].
- une représentation graphique de l'architecture logicielle obtenue, suivant 5 niveaux. Sur ce graphe, nous interprétons les liens reliant les UD de la hiérarchie à travers deux classes: la classe des liens de décomposition (de données ou d'opérations) et la classe des opérations exportées.

L'architecture logicielle présentée offre une image réelle du système à travers ses aspects statique et dynamique et permet de définir un cadre conceptuel pour la réalisation du système.

Références

- [1] J.R.Abrial, Une approche formelle du développement des logiciels, Génie Logiciel & Systèmes Experts N°11, Mars 1988.
- [2] M.Desbois, L.Marie & J.P.Martin, La technique de la réparation automobile, Tome1 Le moteur-Mise au point et contrôles, Essais et mesures, Ed.Foucher 1984.
- [3] M.Menardon, La réparation automobile, Chotard et associés Editeurs 1979.
- [4] M.Noureddine, SACREL: Un Système d'Aide à la Conception et à la Réalisation de Logiciels, Conception et réalisation du noyau, Thèse de Magister, USTO, Décembre 1990.
- [5] M.Noureddine, Contribution à la réalisation d'un système automatisé de détection des pannes moteur, 4^{ème} séminaire national de Mécanique, ENSET Oran, Avril 1995.