

Application d'une métrique pour la conception mécatronique et son implémentation dans SysML

A. WARNIEZ^a, O. PENAS^a, T. SORIANO^b

a. LISMMA, Supméca Paris, 3 rue F. Hainaut, 93407 St Ouen Cedex

b. LISMMA, Supméca Toulon, Maison des technologies, Place Georges Pompidou 83000 Toulon

Résumé :

La conception de systèmes mécatroniques fait apparaître deux spécificités fondamentales : l'approche multi-niveau et l'aspect multi-domaine. Ces deux points impliquent que de nombreux modèles vont être développés, échangés, modifiés, hiérarchisés, sélectionnés, avant l'aboutissement à un prototype virtuel réaliste du système à concevoir. Or, il est souvent difficile pour les concepteurs de choisir, à chaque étape du processus de conception des produits mécatroniques, les meilleurs modèles à valider. Pour répondre à ce besoin, des objectifs « haut-niveau » (robustesse, coût, fiabilité, intégration...), doivent d'abord être définis, puis traduits en critères mathématiques. Puis, les métriques permettront de quantifier (au travers de ces critères) les performances des modèles par rapport aux objectifs fixés, pour justifier les choix des concepteurs, capitaliser un savoir-faire et avoir une traçabilité de la conception. Notre première approche est de s'appuyer sur l'ingénierie système basée sur les modèles, en adéquation avec les besoins de conception mécatronique, pour montrer la faisabilité d'un tel processus, au travers d'un critère choisi, appliqué à un modèle SysML d'un système mécatronique : le groupe moteur propulseur électrique (GMPE).

Abstract:

Design of mechatronic systems reveals two fundamental features: multi-level approach and multi-domain aspect. These both points imply that many models will be developed, exchanged, altered, prioritized, selected, before obtaining a realistic virtual prototype. However, it is often difficult for designers to choose at each step of the design of mechatronic products, the best models to validate. To meet this need, objectives "high level" (robustness, cost, durability, integration ...) must first be defined and translated into mathematical criteria. Then, the metrics will quantify (through the criteria) model performance with regards to fixed targets, to justify the choice of designers, capitalize know-how and have traceability of the design. Our first approach is to rely on engineering system based on models, in accordance with the needs of mechatronic design, to show the feasibility of such a process, through a selected criterion applied to a SysML model of a mechatronic system: the electrical powertrain.

Mots clefs : modèle, conception mécatronique, métrique, critère, validation, SysML

1. Introduction

La conception d'un système est la transformation des besoins clients en une solution optimisée. En effet, l'objectif du concepteur est de fournir un produit le plus conforme possible aux besoins exprimés dans le cahier des charges fourni par le client, et sur l'ensemble de son cycle de vie. Il existe différents cycles de conception mais le plus utilisé est celui du cycle en V. Ce cycle est une démarche méthodique et structurée qui permet d'aider le concepteur tout au long de son projet, depuis le cahier des charges jusqu'aux tests de réception du système. Le cycle en V est constitué d'une phase descendante et d'une phase montante. Nous nous intéresserons dans notre étude à la phase descendante du cycle en V (analyse du besoin, niveau fonctionnel, niveau logique/organique et niveau physique/conception détaillée). L'utilisation des métriques permet de faciliter le travail du concepteur de systèmes mécatroniques. En effet, la conception des produits mécatroniques est particulièrement

longue et difficile, du fait d'une part de leur forte intégration fonctionnelle, de leurs aspects multi-domaine et multi-physique et d'autre part des couplages qui en découlent, qui viennent se rajouter à l'aspect multi-niveau que nous venons de décrire. Les systèmes mécatroniques peuvent donc être dans ce sens considérés comme des systèmes complexes. L'interaction entre plusieurs domaines (la mécanique, l'électronique, l'automatique et l'informatique...) et l'ingénierie associée touchent des métiers multiples et variés, et favorisent donc la création de nombreux modèles, particulièrement dans la phase descendante du cycle en V, qui est une phase de création. En effet, il n'existe pas une conception unique idéale pour concevoir le meilleur système mécatronique. La problématique qui se pose est donc de pouvoir sélectionner le ou les bons modèles à valider, tout au long du processus de conception et donc d'aider le concepteur dans ses choix pour augmenter la pertinence, l'efficacité et la qualité des modèles par rapport aux objectifs de conception fixés initialement.

Même si certains auteurs ont déjà travaillé sur la réalisation d'une maquette globale virtuelle, limitant ainsi le problème de multiplicité des modèles en choisissant l'unification [1] [2], de notre côté, nous avons orienté notre démarche vers la capacité à "mesurer la qualité d'un modèle" pour faciliter le choix du concepteur, soumis à de nombreux modèles et outils, tout au long de la conception. Pour répondre à ce besoin, des objectifs "haut-niveau" (coût, robustesse, sûreté [3], fiabilité [4], esthétique, flexibilité [5], facilité de fabrication,...) doivent d'abord être définis, par exemple, Guillerme et al. ont montré l'implémentation de la sûreté dans leur modélisation SysML [3]. Puis ces objectifs doivent être traduits en critères mathématiques [6] [7]. Ensuite, il s'agit de trouver un moyen de mesurer et contrôler l'adéquation des modèles avec ces objectifs : ce sont le rôle des métriques [8] [9]. En effet, elles permettent de quantifier (au travers de ces critères) les performances des modèles par rapport aux objectifs fixés, pour justifier les choix des concepteurs, capitaliser un savoir-faire et avoir une traçabilité. Ainsi, elles aideront le concepteur à prendre des décisions, à détecter et vérifier les défauts tôt dans le cycle de conception.

Historiquement, les métriques sont apparues dans le domaine de l'informatique [10] et sont de plus en plus développées à l'heure actuelle. Aujourd'hui, les métriques existent à tous les niveaux de conception : par exemple au niveau logique sous Simulink, avec l'outil Simulink Design Verifier [11], et dans de nombreux domaines, quel que soit le secteur d'application. Ainsi, dans le domaine du transport, Alstom utilise des critères d'éco-conception comme la consommation en kWh/km/passager, le poids par passager, le taux de recyclabilité, le taux de substances dangereuses, le taux de matériel renouvelable, le niveau de bruit sonore et la concentration d'émission de particules [12], pour valider la conception de leurs produits. Dans le domaine de l'automobile, A. G. MHALACHE expose dans sa thèse, une méthodologie pour évaluer et estimer la faisabilité des systèmes mécatroniques par la mesure de la fiabilité, la disponibilité, la maintenance et la sécurité [13].

Pour la conception mécatronique, certains chercheurs ont déjà formalisé des critères pour les métriques [14] [15]. Ainsi, S. TURKI [15] a défini un certain nombre de critères (critère de centralisation, l'indicateur de couplage syntaxique, l'indicateur de couplage effectif et pour finir l'indicateur d'intégration mécatronique et l'indicateur d'autonomie d'un bloc) pour évaluer le haut-niveau d'intégration des systèmes mécatroniques. Considérant la démarche de conception d'un système mécatronique comme une déclinaison de l'ingénierie des systèmes avec ses spécificités, il a établi ses critères, en vue d'une utilisation sur des modèles SysML. En effet, le langage SysML (**S**ystems **M**odeling **L**anguage) [16], développé pour l'ingénierie système, permet de modéliser un système sous forme de diagrammes, quelque soit le domaine, de la définition des exigences à l'architecture organique et ainsi fournir un même jeu de paramètres à toutes les équipes techniques intervenant dans la conception.

2. Démarche issue de l'ingénierie basée sur les modèles

Initialement, les grands projets utilisaient traditionnellement une approche d'ingénierie basée sur la documentation. Cette approche générait une documentation de spécifications et de conception en papier qui était ensuite échangée entre les clients, les ingénieurs, les concepteurs, les développeurs et les testeurs. Le but était d'avoir un système conforme à la documentation [17]. La vérification de la cohérence et des interactions entre ces documents entre eux se faisait manuellement et généralement trop tardivement dans le cycle de conception. Cette approche avait des limites au niveau du contenu,

des relations entre les exigences ; la conception et les informations techniques et de tests étaient difficilement accessibles car ces informations étaient dispersées dans plusieurs documents. Cela rendait difficile la traçabilité et l'évaluation de la conception mais aussi la mise-à-jour et la réutilisation des modèles. C'est pour cela, que l'ingénierie système basée sur les modèles a vu le jour et notamment le langage UML, puis le langage SysML. En effet, avec l'ingénierie basée sur les modèles toutes les informations de conception sont maintenant contenues dans un modèle de données. Le modèle contient des éléments qui représentent les exigences, les modèles de conception, les cas d'essais et leurs interconnexions. Le but de l'ingénierie système basée sur les modèles est d'obtenir une modélisation cohérente du système en cours de développement. Les informations précédemment saisies dans la documentation sont maintenant intégrées dans le modèle de données. La documentation est automatiquement générée par le modèle : il est plus facile de voir toutes les informations et les relations du système. De plus, le temps et le prix de la réalisation et de la maintenance de la documentation sont significativement réduits. L'ingénierie système basée sur les modèles assure donc une cohérence et une meilleure communication entre les équipes de conception.

Un modèle SysML permet de représenter les exigences du système, de décrire les flux et les allocations (propriétés), l'interconnexion des parties (structure) du système, et les éléments et équations physiques (comportement du système). Les outils, libres ou à évaluation gratuite, proposant l'implémentation du langage SysML sont notamment ARTISAN STUDIO [18], TOPCASED [19], PAPYRUS [20]. Les relations de traçabilité de SysML semblent fort intéressantes surtout dans le contexte de la conception mécatronique où les interactions entre les différents domaines et les divers composants multi-physiques doivent être impérativement tracées.

3. Expérimentation

En ingénierie, des critères peuvent être établis pour estimer si un modèle satisfait ses objectifs de modélisation. Ces critères et les outils utilisés ou développés pour les métriques associées [10], aideront les concepteurs à développer des modèles en conformité avec les objectifs fixés.

Dans cet article, nous nous appuyerons sur un critère proposé par S. TURKI [15], afin de tester la faisabilité de l'implémentation des métriques dans un modèle SysML, pour la conception d'un produit mécatronique (FIG. 1) : le critère de centralisation.

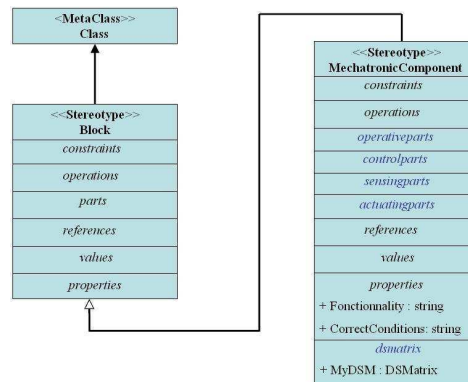


FIG. 1 – Description du composant mécatronique par [15]

Le critère de centralisation CC d'un composant mécatronique MC_i est défini comme étant le nombre de ses parties « contrôle » divisé par la somme du nombre de ses parties opératives et de ses composants mécatroniques imbriqués dans le système mécatronique :

$$CC_i = \frac{n_i}{m_i + NbMC_i} \quad (1)$$

n_i : nombre de parties opératives (et d'actionnement) du composant MC_i .

m_i : nombre de parties « contrôle » (et de capture) de MC_i .

$NbMC_i$: nombre de composants mécatroniques contenus dans MC_i .

Si ce critère décroît, le composant mécatronique devient plus distribué. À l'inverse lorsqu'il croît, le composant devient plus centralisé. Le choix de conception revient toujours au concepteur. S'il

privilégie la sécurité, il choisira une architecture plus centralisée. Au contraire, s'il privilégie l'autonomie, la performance, il choisira une architecture plus distribuée.

Cette métrique a été testée sur un modèle SysML du groupe motopropulseur électrique (GMPE). Le GMPE, correspondant à l'ensemble des éléments participant à la motricité d'un véhicule, a été modélisé avec le logiciel Artisan Studio, en collaboration avec la société EIRIS, dans le cadre du projet O2M (Outils de Modélisation Mécatronique), labellisé par le pôle Mov'éo. Nous avons utilisé le critère de centralisation sur un diagramme de bloc interne (IBD), qui décrit la structure interne du système en termes de parties, ports et connecteurs [17]. Il fournit le contexte de tous les éléments du diagramme et montre principalement les relations entre les éléments de même niveau. Dans notre modélisation, il est utilisé pour décrire l'architecture logique du système, avec ses flux et ses ports (FIG.2). Dans le cas du GMPE, il montre l'interaction des éléments suivants : le traducteur de consigne du conducteur (qui correspondra au composant pédale) qui permet de traduire l'accélération ou le freinage, un élément de contrôle (qui correspondra au composant calculateur), une alimentation (batterie), un modulateur de la puissance électrique (rôle de l'onduleur), un convertisseur pour transformer la puissance électrique en puissance mécanique (rôle de la machine électrique) et une transmission pour adapter la puissance mécanique qui sera délivrée sur les roues (rôle du réducteur).

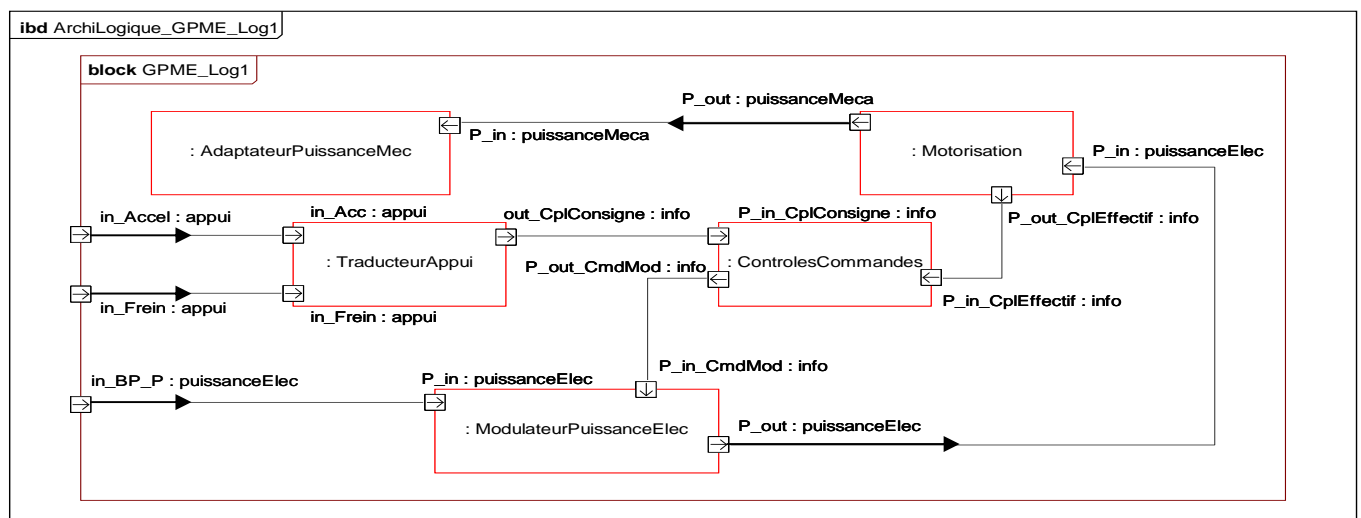


FIG. 2 – Diagramme IBD de l'architecture logique du GMPE.

Sur ce diagramme, on peut identifier :

- 1 partie commande – Bloc : ControlesCommandes
- 4 parties opératives – Bloc : TraducteurAppui
- Bloc : ModulateurPuissanceElec
- Bloc : Motorisation
- Bloc : AdaptateurPuissanceMec

Le critère de centralisation sur ce modèle est donc égal à

$$CC_i = \frac{4}{1+0} = 4 \quad (2)$$

Puisqu'il n'y pas de composants mécatroniques imbriqués dans ce système (sous-bloc). On différencie par les flux énergétiques les parties opératives. Lorsque ce résultat tend vers 1, le système est complètement distribué. Il n'y a pas de composants mécatroniques imbriqués et il y a autant de partie opératives que de parties commandes. La limite haute reste encore à définir par étude et expérimentation

4. Outils envisagés pour l'automatisation

Pour pouvoir exécuter de façon automatique les métriques sur les modèles, il faut tout d'abord trouver un ou plusieurs outil(s) pour extraire les informations du modèle contenant les éléments qui nous intéressent : dans notre exemple, il s'agit des blocs et des liens, puis de trier/sélectionner parmi les lignes de code extraites, celles qui vont permettre de calculer le critère de la métrique (nombre de

blocs, la classe des objets (« control part » et « operative part »), puis d'implémenter la métrique pour obtenir le résultat recherché. Nous avons donc identifié quelques pistes : des outils utilisant Python [21], puisque des études au laboratoire montrent que l'on peut programmer en Python, pour extraire des données de modèles de conception, et par ailleurs des outils implémentant SysML (Artisan Studio, Topcased, Eclipse...) commencent à mettre en place les extractions sous XML [18] [22].

En effet, il s'agit non seulement d'exporter les données du calcul du critère mais aussi de choisir aussi un format du fichier d'extraction, qui soit compatible avec le modèle original et l'outil d'extraction choisi, afin d'automatiser ce calcul. Ainsi, pour les modèles SysML, nous nous sommes naturellement orientés vers le format XMI/XML. En effet, le langage UML s'appuie sur le standard XMI (eXtensible Markup Language) pour le stockage et l'échange de ses modèles, ce qui permet d'une part pour un utilisateur de passer d'un outil de modélisation à un autre sans que le modèle soit modifié et d'autre part de faciliter la réalisation d'outils de génération automatique de code. Puisque SysML est défini comme une extension métamodèle UML, les modèles SysML peuvent être facilement échangés en utilisant un schéma XMI/XML [22].

Par ailleurs, une fois ces métriques réalisées, il peut être intéressant de proposer au concepteur de choisir le modèle le plus en adéquation avec les objectifs initiaux fixés, et de retourner l'information au logiciel de modélisation, pour valider le modèle correspondant pour la maquette numérique définitive. Ainsi, la plate-forme SMART, initialement développée pour les fichiers de configuration, permet de convertir des fichiers (extraits de modèles par exemple), dont le format est XML, dans un format structuré à définir, de le modifier, puis de le reconverter dans un fichier au format initial, pour le réintégrer à l'outil d'où a été extrait le fichier initial [23].

Par ailleurs, il ne faut pas oublier que les métriques sont là pour évaluer un modèle ou une solution d'architecture. Pour aider le concepteur à faire le bon choix, les modèles (et/ou les solutions d'architectures) peuvent être comparés entre eux pour un même cahier des charges donné. Aussi par la suite, le critère sera appliqué à un système plus centralisé tel que le système moteur-roue, comme "l'Active Wheel" de Michelin [24] (FIG. 3). En effet, le moteur-roue supprime la transmission mécanique et la boîte de vitesse (supprimant 2 composants), en facilitant le contrôle du couple et de la vitesse.

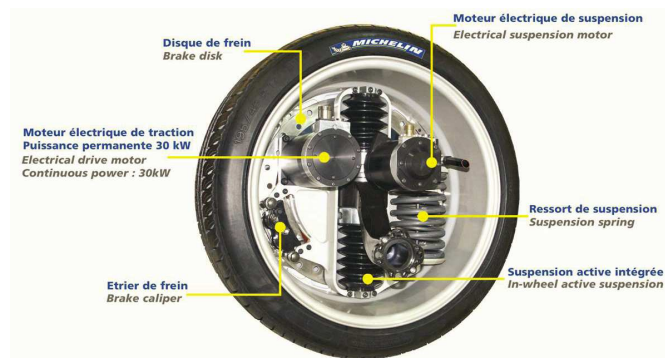


FIG. 3 – Moteur-roue Active Wheel® de Michelin.

5. Conclusions

Cet article traite de la conception mécatronique et plus particulièrement de la validation de modèles pour l'ingénierie. La conception mécatronique est complexe (couplages multi-domaines et multi-physiques), et un des problèmes est de sélectionner le ou les bons modèles pour satisfaire les objectifs du concepteur tout en garantissant les exigences du client. Le but de notre étude est de proposer des critères pour mesurer la qualité des modèles de conception mécatronique afin d'aider le concepteur dans ses choix. Pour cela, des critères spécifiques pour évaluer le haut-niveau d'intégration des systèmes mécatroniques ont été identifiés principalement pour l'ingénierie basée sur les modèles. Le critère de centralisation a été expérimenté sur un modèle SysML d'un groupe motopropulseur électrique. Cette première approche commence par l'expérimentation d'une première métrique qui sera la base de notre étude future. En effet, il reste encore d'autres métriques, que nous avons identifiées et qu'il nous faut encore formaliser et valider, pour cerner toute la complexité des systèmes

mécatroniques, et ce quel que soit le niveau de conception considéré de la partie descendante du cycle en V.

References

- [1]. **GOURIET, P.** *Involving AUTOSAR Rules for Mechatronic System Design*. CSDM : Springer, 2010. 978-3-642-15653-3.
- [2]. **PLATEAUX, R., et al.** Integrated Design Methodology of a Mechatronic System. Septembre-Octobre 2010, Vol. 11, 5, pp. 401 - 406.
- [3]. **GUILLERM, R., DEMMOU, H. et N., SADOU.** *Information Model for Model Driven Safety Requirements Management of Complex Systems*. CSDM : Springer, 2010. 978-3-642-15653-3.
- [4]. **ZHONG, X., ICHHOU, M. et SAIDI, A.** Reliability assessment of complex mechatronic systems using a modified nonparametric belief propagation algorithm. 2010, Vol. 95, pp. 1174-1185.
- [5]. **MOSES, J.** *Flexibility and Its Relation to Complexity and Architecture*. CSDM : Springer, 2010. 978-3-642-15653-3.
- [6]. **GIAKOUMAKIS, V., et al.** *Optimal Technological Architecture Evolutions of Information Systems*. CSDM : Springer, 2010. 978-3-642-15653-3.
- [7]. **VINTR, Z. et VINTR, M.** *Contribution to Rational Determination of Warranty Parameters for a New Product*. CSDM : Springer, 2010. 978-3-624-15653-3.
- [8]. **GILB, T.** Scales of Measure. *A Handbook for Systems Engineering, Requirements Engineering and Software Engineering Using Planguage*. s.l. : Butterworth-Heinemann Ltd, 2005, 5. http://www.gilb.com/tiki-download_file.php?fileId=26.
- [9]. **GRAY, A.R. et MACDONELL, S.G.** A comparison of techniques for developing predictive models of software metrics. 1997, Vol. 39, pp. 425-437.
- [10]. **PRINTZ, J.** *A Natural Measure for Denoting Software System Complexity*. CSDM : Springer, 2010. 978-3-642-5-15653-3.
- [11]. **ETIENNE J.F., FECHTER S., JUPPEAUX E.** *Using Simulink Design Verifier for Proving Behavioral Properties on a Complex Safety Critical System in the Ground Transportation Domain*. CSDM : Springer, 2010. 978-3-642-15653-3.
- [12]. **ANDRIES, V.** *ECODESIGN in ALSTOM TRANSPORT : Environmental impacts and aspects integration in Rolling stock design*. Saint Ouen : Innovaxiom - Mécatronique innovante, 16/11/2010.
- [13]. **MIHALACHE, A. G.** *Modélisation et évaluation de la fiabilité des systèmes mécatroniques : application sur système embarqué*. Université d'Angers : s.n., 2007.
- [14]. **HEHENBERGER, P., et al.** Hierarchical design models in the mechatronic product development process of synchronous machines. 2010, Vol. 20, 8, pp. 864-875.
- [15]. **TURKI, S.** *Ingénierie système guidée par les modèles: Application du standard IEEE 15288, de l'architecture MDA et du langage SysML à la conception des systèmes mécatroniques*. LISMMA Toulon : Université de Toulon, oct. 2008.
- [16]. **SysML.** SYStems Modeling Language (SysML) specification. Version 1.0 , Draft. OMG document ad/2006-03-01. *SysML*. [En ligne] 2006. <http://www.sysml.org/> <http://www.omgsysml.org/>.
- [17]. **FRIDENTHAL et STEINER, MOORE.** *A practical guide to SysML: Systems Model Language*. s.l. : Morgan Kaufmann, 2008. 9780123743794.
- [18]. ATEGO, Artisan Studio page d'accueil. <http://www.atego.com/products/artisan-studio/>. [En ligne]
- [19]. TOPCASED, The Open-Source Toolkit for Critical system. <http://www.topcased.org>. [En ligne]
- [20]. Papyrus Homepage, an Open Source Tool for Graphical UML2 Modeling. <http://www.papyrusuml.org>. [En ligne]
- [21]. Python Programming Language – Official Website. www.python.org. [En ligne]
- [22]. *OMG Systems Modeling Language*. [En ligne] <http://www.omgsysml.org/>.
- [23]. **PAULINO, H., MARTINS, A., LOURENÇO, J. et DURO, N.** *SmART : An application Reconfiguration Framework*. CSDM : Springer, 2010. 978-3-6242-15653-3.
- [24]. Active Wheel. www.michelin.com. [En ligne] 2008. http://fr.wikipedia.org/wiki/Active_Wheel.