UNIVERSITÉ PARIS-SUD

ÉCOLE DOCTORALE INFORMATIQUE PARIS-SUD
Laboratoire de CEA Tech LIST

DISCIPLINE : Informatique

SYNTHÈSE EN FRANÇAIS

THÈSE DE DOCTORAT

soutenue le 07/07/2014

par

Ernest Wozniak

Synthèse Basée sur les Modèles d'Architectures Automobiles Temps Réel Distribuées

Directeur de thèse:	Dr. Sébastien Gérard	CEA Tech LIST
Co-directeur de thèse:	Dr. Chokri Mraidha	CEA Tech LIST

Composition du jury

Président du jury :	Burkhart Wolff	Professeur (Université Paris-Sud, CNRS)
Rapporteur :	Maryline Chetto	Professeur (Université de Nantes, IRCCyN Research Institute)
Rapporteur :	Martin Törngren	Professeur (KTH Royal Institute of Technology)
Examinateur :	Claire Pagetti	Ingénieur-Chercheur (ONERA)
Examinateur :	Manfred Broy	Professeur (Technische Universität München)

Bibliographie

[1] ISO/IEC/(IEEE), “ISO/IEC/IEEE 42010:2011 : Systems and software engineering - Architecture description,” 11 2011.

[2] T. Dittel and H.-J. Aryus, “How to "survive" a safety case according to iso 26262,” in *Computer Safety, Reliability, and Security*, ser. Lecture Notes in Computer Science, E. Schoitsch, Ed. Springer Berlin Heidelberg, 2010, vol. 6351, pp. 97–111.

[3] M. Broy, “Challenges in automotive software engineering,” in *Proceedings of the 28th international conference on Software engineering*, ser. ICSE ’06. New York, NY, USA: ACM, 2006, pp. 33–42. [Online]. Available: http://doi.acm.org/10.1145/-1134285.1134292

[4] M. Broy, I. Krüger, A. Pretschner, and C. Salzmann, “Engineering Automotive Software,” *Proceedings of the IEEE*, vol. 95, no. 2, 2007.

[5] AUTOSAR, http://www.autosar.org/.

[6] *EAST-ADL Domain Model Specification V2.1.11*, 05 2013.

[7] D. Ku and G. Micheli, “EnglishDesign space exploration,” in *EnglishHigh Level Synthesis of ASICs under Timing and Synchronization Constraints*, ser. The Springer International Series in Engineering and Computer Science. Springer US, 1992, vol. 177, pp. 83–111. [Online]. Available: http://dx.doi.org/10.1007/978-1-4757-2117-1_5

[8] *AUTOSAR Specification of Timing Extensions*, AUTOSAR Std. 1.2.0. [Online]. Available: http://www.autosar.org/download/R4.0/-AUTOSAR_TPS_TimingExtensions.pdf

[9] S. Anssi, “Methodology for model-based timing analysis process for automotive systems,” Ph.D. dissertation, l’Université Paris-Sud, 2011.

[10] *Unified Modeling Language Superstructure v2.3*, OMG Std.

[11] SysML, http://www.sysml.org/.

[12] *UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded Systems, Version 1.0, formal/2009-11-02*, OMG, November 2009. [Online]. Available: http://-www.omgmarte.org/

[13] OSEK/VDX, http://www.osek-vdx.org/.

[14] *Specification of Operating System V5.2.0*, 10 2013.

[15] H. Kopetz, *Real-Time Systems: Design Principles for Distributed Embedded Applications*, 1st ed. Norwell, MA, USA: Kluwer Academic Publishers, 1997.

[16] ——, "Event-triggered versus time-triggered real-time systems," in *Operating Systems of the 90s and Beyond*, ser. Lecture Notes in Computer Science, A. Karshmer and J. Nehmer, Eds. Springer Berlin Heidelberg, 1991, vol. 563, pp. 86–101. [Online]. Available: http://dx.doi.org/10.1007/BFb0024530

[17] H. Kopetz and G. Bauer, "The time-triggered architecture," *Proceedings of the IEEE*, vol. 91, no. 1, pp. 112–126, 2003.

[18] H. Kopetz and G. Grunsteidl, "Ttp-a protocol for fault-tolerant real-time systems," *Computer*, vol. 27, no. 1, pp. 14–23, 1994.

[19] O. Scheickl, "Timing constraints in distributed development of automotive real-time systems," Ph.D. dissertation, Technische Universität München für Informatik, 2011.

[20] U.S. Government Department of Defense, "The DoDAF Architecture Framework Version 2.02." 2009.

[21] MODAF partners, "MOD Architectural Framework Technical Handbook, Version 1.0," August 2005.

[22] EAST-EEA, http://www.itea2.org/public/project_leaflets/EAST-EEA_results_oct-04.pdf/.

[23] ATESST, http://www.atesst.org/.

[24] D. Chen, H. Lönn, F. Törner, and H. Blom, "Advancing traffic efficiency and safety through software technology (atesst) deliverable d2.2.2," Tech. Rep., January 2008. [Online]. Available: http:/www.atesst.org/

[25] M. Weber and H. Lönn, "Methodology guideline when using east-adl2," Tech. Rep. Deliverable D5.1.1, June 2010. [Online]. Available: http://www.atesst.org/home/-liblocal/docs/ATESST2_Deliverable_D5.1.1_V1.1.pdf

[26] *AUTOSAR Methodology*, AUTOSAR Std. [Online]. Available: http://-www.autosar.org/download/AUTOSARMethodology.pdf

[27] T. N. Qureshi, D.-J. Chen, H. Lönn, and M. Törngren, "From east-adl to autosar software architecture: A mapping scheme," in *ECSA*, 2011, pp. 328–335.

[28] *International Standards Organization, ISO/DIS 26262:2009 - Draft International Standard Road Vehicles - Functional Safety"*, http://www.iso.org, Std.

[29] *TIMMO-2-USE Timing Model - Tools, algorithms, languages, methodology, USE cases*, TIMMO-2-USE Partners, October 2012. [Online]. Available: http://-www.timmo-2-use.org/

[30] *dSpace*, http://www.dspaceinc.com/.

[31] Vector, http://www.vector.com/.

[32] S. (SymtaVision), http://www.symtavision.com/symtas.html.

[33] SynDEx, http://www.syndex.org/.

[34] K. Tindell and J. Clark, "Holistic schedulability for distributed hard real-time systems," *Microprocessing & Microprogramming*, vol. 40, pp. 117–134, 1994.

[35] J. C. Palencia and M. G. Harbour, "Schedulability analysis for tasks with static and dynamic offsets," in *Proceedings of the 19th IEEE Real-Time Systems Symposium*, 1998, p. 26.

[36] Q. Zhu, Y. Yang, E. Scholte, M. Di Natale, and A. Sangiovanni-Vincentelli, "Optimizing extensibility in hard real-time distributed systems," in *Proceedings of the 15th IEEE Real-Time and Embedded Technology and Applications Symposium*. IEEE, 2009, pp. 275–284.

[37] L. Sha, R. Rajkumar, and J. Lehoczky, "Priority inheritance protocols: An approach to real-time synchronization," *Computers, IEEE Transactions on*, vol. 39, no. 9, pp. 1175–1185, 1990.

[38] *The Mathworks Simulink and StateFlow User's Manuals*. [Online]. Available: http://-www.mathworks.com

[39] H. Zeng and M. D. Natale, "Efficient implementation of autosar components with minimal memory usage," in *SIES*, 2012, pp. 130–137.

[40] B. Buhnova, L. Grunske, A. Koziolek, and I. Meedeniya, "Software architecture optimization methods: A systematic literature review," *IEEE Transaction on Software Engineering*, 2012.

[41] M. Stigge, P. Ekberg, N. Guan, and W. Yi, "The digraph real-time task model," in *Real-Time and Embedded Technology and Applications Symposium (RTAS), 2011 17th IEEE*, April 2011, pp. 71–80.

[42] P. Pop, P. Eles, Z. Peng, and T. Pop, "Analysis and optimization of distributed real-time embedded systems," *ACM Transactions on Design Automation of Electronic Systems*, vol. 11, no. 3, pp. 593–625, 2006.

[43] T. Pop, P. Eles, and Z. Peng, "Design optimization of mixed time/event-triggered distributed embedded systems," in *Proc. of the First IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, New York, NY, USA, 2003.

[44] X. He, Z. Gu, and Y. Zhu, "Task allocation and optimization of distributed embedded systems with simulated annealing and geometric programming," *The Computer Journal*, vol. 53, no. 7, pp. 1071–1091, 2010.

[45] I. Bate and P. Emberson, "Incorporating scenarios and heuristics to improve flexibility in real-time embedded systems," in *Proceedings of the 12th IEEE Real-Time and Embedded Technology and Applications Symposium*, 2006, pp. 221–230.

[46] A. Hamann, R. Racu, and R. Ernst, "Multi-dimensional robustness optimization in heterogeneous distributed embedded systems," in *Proceedings of the 13th IEEE Real Time and Embedded Technology and Applications Symposium*, April 2007.

[47] E. Azketa, J. Uribe, J. Gutierrez, M. Marcos, and L. Almeida, "Permutational genetic algorithm for the optimized assignment of priorities to tasks and messages in distributed real-time systems," in *Proceedings of the 10th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, 2011, pp. 958–965.

[48] S. Kugele, W. Haberl, M. Tautschnig, and M. Wechs, "Optimizing automatic deployment using non-functional requirement annotations," *Leveraging Applications of Formal Methods, Verification and Validation*, pp. 400–414, 2009.

[49] M. Richard, P. Richard, and F. Cottet, "Allocating and scheduling tasks in multiple fieldbus real-time systems," in *Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation*, vol. 1. IEEE, 2003, pp. 137–144.

[50] C. Bartolini, G. Lipari, and M. Di Natale, "From functional blocks to the synthesis of the architectural model in embedded real-time applications," in *Proc. 11th IEEE Real Time and Embedded Technology and Applications Symposium*, 2005, pp. 458–467.

[51] M. Saksena, P. Karvelas, and Y. Wang, "Automatic synthesis of multi-tasking implementations from real-time object-oriented models," in *Proceedings of 3rd IEEE International Symposium on Object-Oriented Real-Time Distributed Computing*, 2000, pp. 360–367.

[52] S. Kodase, S. Wang, and K. Shin, "Transforming structural model to runtime model of embedded software with real-time constraints," in *Proceedings of the conference on Design, Automation and Test in Europe*, 2003, pp. 170–175.

[53] Q. Zhu, H. Zeng, W. Zheng, M. D. Natale, and A. Sangiovanni-Vincentelli, "Optimization of task allocation and priority assignment in hard real-time distributed systems," *ACM Transactions on Embedded Computing Systems*, vol. 11, no. 4, pp. 85:1–85:30, 2012.

[54] H. Zeng, M. D. Natale, and Q. Zhu, "Optimizing stack memory requirements for real-time embedded applications," in *ETFA*, 2012, pp. 1–8.

[55] A. Ferrari, M. D. Natale, G. Gentile, G. Reggiani, and P. Gai, "Time and memory tradeoffs in the implementation of autosar components," in *DATE*, 2009, pp. 864–869.

[56] M. Zhang and Z. Gu, "Optimization issues in mapping autosar components to distributed multithreaded implementations," in *International Symposium on Rapid System Prototyping*, 2011, pp. 23–29.

[57] L. Davis, Ed., *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, 1991.

[58] B. L. Miller, B. L. Miller, D. E. Goldberg, and D. E. Goldberg, "Genetic algorithms, tournament selection, and the effects of noise," *Complex Systems*, vol. 9, pp. 193–212, 1995.

[59] A. Mehiaoui, E. Wozniak, S. T. Piergiovanni, C. Mraidha, M. D. Natale, H. Zeng, J.-P. Babau, L. Lemarchand, and S. Gérard, "A two-step optimization technique for functions placement, partitioning, and priority assignment in distributed systems," in *SIGPLAN/SIGBED Conference on Languages, Compilers and Tools for Embedded Systems LCTES*, 2013, pp. 121–132.

[60] S. Anssi, S. Tucci-Piergiovanni, S. Kuntz, S. Gerard, and F. Terrier, "Enabling scheduling analysis for autosar systems," *Object-Oriented Real-Time Distributed Computing, IEEE International Symposium on*, vol. 0, pp. 152–159, 2011.

[61] C. Mraidha, S. Tucci-Piergiovanni, and S. Gerard, "Optimum: a marte-based methodology for schedulability analysis at early design stages," *ACM SIGSOFT Software Engineering Notes*, vol. 36, no. 1, pp. 1–8, 2011.

[62] E. Wozniak, S. Tucci-Piergiovanni, C. Mraidha, and S. Gerard, "An integrated approach for modeling, analysis and optimization of systems whose design follows the east-adl2/autosar methodology," *SAE International Journal of Passenger Cars - Electronic and Electrical Systems*, vol. 6(1), no. 276-286, 2013.

[63] A. Mehiaoui, S. T. Piergiovanni, J.-P. Babau, and L. Lemarchand, "Optimizing the deployment of distributed real-time embedded applications," in *RTCSA*, 2012, pp. 400–403.

[64] J. Rox, K. Schmidt, A. Winter, T. Spengler, and R. Ernst, "Estimating and mitigating design risk in a flexible distributed design process," *IEEE Embedded Systems Letters*, vol. 2, no. 2, pp. 35–38, 2010.

[65] *ALL TIMES: D2.23.2 Final prototype of integrated system-level verification methodology*, ALL TIMES Partners, August 2010. [Online]. Available: http://-www.mrtc.mdh.se/projects/all-times/

[66] M. Di Natale and J. A. Stankovic, "Dynamic end-to-end guarantees in distributed real time systems," in *Proceedings of the IEEE Real-Time Systems Symposium*, 1994, pp. 216–227.

[67] R. Gerber, S. Hong, and M. Saksena, "Guaranteeing real-time requirements with resource-based calibration of periodic processe," in *IEEE Transactions on Software Engineering*, vol. 21, July 1995, pp. 579–592.

[68] N. Serreli and E. Bini, "Deadline assignment for component-based analysis of real-time transactions," in *2nd Workshop on Compositional Real-Time Systems, Washington, DC, USA*, 2009.

[69] S. Hong, T. Chantem, and X. S. Hu, "Meeting end-to-end deadlines through distributed local deadline assignment," in *Proceedings of the IEEE Real-Time Systems Symposium*, 2011.

[70] P. Jayachandran and T. Abdelzaher, "Delay composition in preemptive and non-preemptive real-time pipelines," in *Real-Time Systems Journal*, vol. 40, 2008, pp. 290–320.

[71] N. Feiertag, K. Richter, and C. Ficek, "On the decomposition of end-to-end timing requirements in distributed partitioned automotive functions," in *SAE World Congress, Detroit, USA*, 2012.

[72] M. Di Natale and A. L. Sangiovanni-Vincentelli, "Moving from federated to integrated architectures in automotive: The role of standards, methods and tools," *Proceedings of the IEEE*, vol. 98, no. 4, pp. 603–620, 2010.

[73] M. G. Dixit, S. Ramesh, and P. Dasgupta, "Time-budgeting: a component based development methodology for real-time embedded systems," 2013.

[74] M. G. Dixit, P. Dasgupta, and S. Ramesh, "Taming the component timing: A cbd methodology for real-time embedded systems," in *DATE*, 2010, pp. 1649–1652.

[75] R. Henia, A. Hamann, M. Jersak, R. Racu, K. Richter, and R. Ernst, "System level performance analysis - the symta/s approach," *Computers and Digital Techniques, IEE Proceedings* -, vol. 152, no. 2, pp. 148–166, Mar 2005.

[76] R. Racu, A. Hamann, and R. Ernst, "A formal approach to multi-dimensional sensitivity analysis of embedded real-time systems," in *18th Euromicro Conference on Real-Time Systems*, 2006.

[77] Simulink, http://www.mathworks.com/products/simulink/.

[78] R. Isermann, J. Schaffnit, and S. Sinsel, "Hardware-in-the-loop simulation for the design and testing of engine-control systems," *Control Engineering Practice*, vol. 7, no. 5, pp. 643 – 653, 1999. [Online]. Available: http://www.sciencedirect.com/-science/article/pii/S0967066198002056

[79] W.-H. Kwon and S.-G. Choi, "Real-time distributed software-in-the-loop simulation for distributed control systems," in *Computer Aided Control System Design, 1999. Proceedings of the 1999 IEEE International Symposium on*, 1999, pp. 115–119.

[80] M. Graphics, http://www.mentor.com/.

[81] SystemDesk, http://www.dspaceinc.com/en/inc/home/products/sw/-system_architecture_software/systemdesk.cfm/.

[82] TargetLink, http://www.dspace.com/en/pub/home/products/sw/pcgs/targetli.cfm.

[83] D. Developer, http://www.vector.com/vi_davinci_developer_en.html/.

[84] V. toolset, http://www.mentor.com/products/vnd/autosar-products/.

[85] BridgePoint, http://www.mentor.com/products/sm/model_development/bridgepoint/.

[86] Artop, https://www.artop.org/.

[87] Metacase, http://www.metacase.com/.

[88] Systemite, http://www.systemite.se/.

[89] PREEVision, http://vector.com/vi_preevision_en.html.

[90] A. Bauer, M. Broy, J. Romberg, B. Schätz, P. Braun, U. Freund, N. Mata, R. Sandner, P. Mai, and D. Ziegenbein, "Das AutoMoDe-Projekt: Modellbasierte Entwicklung softwareintensiver Systeme im Automobil," *Computer Science – Research and Development*, vol. 22, no. 1, pp. 45–57, 2007.

[91] F. Hölzl and M. Feilkas, "Autofocus 3: a scientific tool prototype for model-based development of component-based, reactive, distributed systems," in *Proceedings of the 2007 International Dagstuhl conference on Model-based engineering of embedded real-time systems*, ser. MBEERTS'07. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 317–322.

[92] S. Voss and B. Schätz, "Deployment and scheduling synthesis for mixed-critical shared-memory applications," in *ECBS*, 2013, pp. 100–109.

[93] M. Broy, M. Gleirscher, S. Merenda, D. Wild, P. Kluge, and W. Krenzer, "Toward a holistic and standardized automotive architecture description," *Computer*, vol. 42, pp. 98–101, 2009.

[94] *Papyrus MDT project webpage: http://www.eclipse.org/modeling/mdt/papyrus/*. [Online]. Available: http://www.eclipse.org/modeling/mdt/papyrus/

[95] Eclipse, http://www.eclipse.org/.

[96] *EAST-ADL UML Profile Specification V2.1.10*, 06 2012.

[97] *Specification of ECU Configuration*, AUTOSAR Std. V3.1.0. [Online]. Available: http://www.autosar.org/download/R4.0/AUTOSAR_TPS_TimingExtensions.pdf

[98] A. Albinet, L. Queran, B. Sanchez, and Y. Tanguy, "Requirement management fro msystem modeling to autosar sw components," in *Embedded Real Time Software and Systems - ERTS*, 2010.

[99] H. D. E. Ortiz, "An integrated model-driven framework for specifying and analyzing non-functional properties of real time systems," Ph.D. dissertation, l'Université d'Evry, 2007.

[100] *EAST-ADL XML Schema*, MAENAD Std. 3.1. [Online]. Available: http://-www.maenad.eu/public/Deliverables/MAENAD_Deliverable_D4.3.1_V3.1.pdf

[101] *Application Interfaces User Guide*, AUTOSAR Std. V1.3.0. [Online]. Available: http://www.autosar.org/download/R4.1/AUTOSAR_EXP_AIUserGuide.pdf

[102] S. Tucci-Piergiovanni, C. Mraidha, E. Wozniak, A. Lanusse, and S. Gerard, "A uml model-based approach for replication assessment of autosar safety-critical applications," *IEEE TrustCom/IEEE ICESS/FCST, International Joint Conference of*, vol. 0, pp. 1176–1187, 2011.

[103] M. Hagner, A. Aniculaesei, and U. Goltz, "Uml-based analysis of power consumption for real-time embedded systems," in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on*, Nov 2011, pp. 1196–1201.

RESUME

Synthèse Basée sur les Modèles d'Architectures Automobiles Temps Réel Distribuées

par

Ernest Wozniak

1. Contexte de la thèse

Les systèmes véhicules d'aujourd'hui sont marqués par un large gamme de solution qui améliorent la performance, la sécurité et le confort de conduit. Des fonctionnalités telles que le système d'auto-parking étaient au-delà de la croyance pour les conducteurs ordinaires, il ya seulement 10 ans. Nous ne pouvons pas toujours cadrer dans notre esprit les voitures autonomes qui ont déjà été prototypé.

Systèmes automobiles sont perçus comme systèmes distribués, embarqués et tems réel. Tout d'abord, fonctionnalités logicielles des véhicules sont distribué sur plusieurs composants matériels embarqués nommées unités de commande électronique (ang. ECU – Electronic Control Unit) ou sur des capteurs/actionneurs. La couche d'application qui s'étend sur des ECU différents est composé des composants logiciels qui peuvent être livré par plusieurs fournisseurs. Le middleware est responsable de la communication entre les composants logiciels distribués. Chaque ECU exécute un système d'exploitation. Tout cela implique une nature distribuée des systèmes automobiles (voir la Figure 1). Deuxièmement leur fonctionnement est serré par les contraintes de temps de différents types, par exemple les contraintes temporelles de bout-en-bout. Par exemple l'ouverture de l'airbag en cas d'accident doit se produire dans les 20 ms. Le dernier est une contrainte de temps réel, dont la violation non seulement affirme le comportement incorrect du système, mais plus important, peut mettre en danger la vie d'humain.

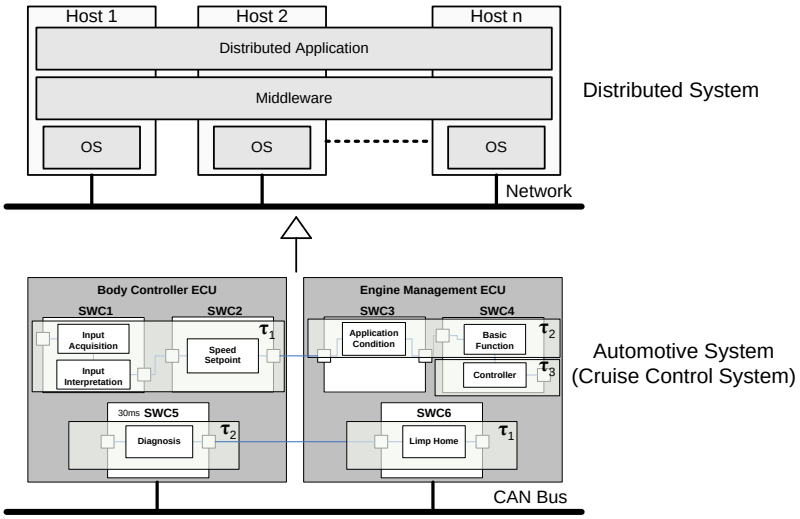

Figure 1. Système Distribué et Système Distribué Automobile

Architectures de systèmes automobiles (en raccourci architectures automobiles) sont des produits très complexes, de haute technologie. Différents facteurs contribuent à leur complexité:

- **Taille:** le nombre de fonctions contrôlées par le logiciel et le matériel est substantiel dans les véhicules d'aujourd'hui. Dans près de 30 ans, la capacité d'un code est passé de 0 à près de 10 Go qui implique des millions de lignes de code.
- **Nature distribuée:** architectures automobiles d'aujourd'hui sont fortement distribués, c.-à-d. fonction atomique de la même fonction du véhicule sont distribué sur plusieurs commande électronique. Le même ECU peut accueillir des fonctions atomiques de différentes fonctions du véhicule. Cela conduit à une meilleure optimisation de l'utilisation des ressources.
- **Les contraintes temps réel:** le correct fonctionnement d'un système de véhicule n'est pas seulement défini par l'absence d'erreurs fonctionnelles, mais aussi par strict respect des contraintes temps réel. Leur existence sert principalement dans les situations critique pour la sécurité, comme le freinage ou pendant un accident lorsque les airbags doivent être activés immédiatement.
- **Exigences de sécurité:** l'aspect de la sécurité joue un rôle important car maintenant ce n'est pas seulement une préoccupation interne d'un OEM (ang. Original Equipment Manufacturer) à fournir des véhicules fiables, mais aussi un sujet pour les réglementations gouvernementales.
- **Exigences contradictoires:** toutes les différentes exigences comme contraintes de temps, la réduction des ressources matérielles pour réduire les coûts, la fourniture de la

sécurité, etc. sont dans le nombreux cas orthogonal. Cela signifie que la satisfaction d'une exigence peut conduire à l'abus ou de la détérioration des autres.

- **Sensible aux changements:** légers changements de désign ou certaines propriétés des objets d'architecture peuvent conduire à une modification radicale des caractéristiques non-fonctionnelles d'architecture. Par exemple, l'augmentation d'un temps d'exécution d'une fonction atomique pourrait conduire à la violation de plusieurs contraintes de temps.

En raison de cette complexité qui a été et est encore en croissance exponentielle (comme présumé pour les 20 prochaines années), de nouvelles stratégies pour la conception des systèmes automobiles doivent être introduites. L'un d'entre eux est l'adoption de l'ingénierie dirigée par les modèles (IDM) pour le développement des systèmes automobiles. Le principe de l'approche IDM consiste à intégrer des abstraits, dans de nombreux cas graphiques modèles pour spécifier les exigences fonctionnelles et non fonctionnelles, et enfin, pour produire un code binaire qui respecte la spécification. Le potentiel de l'IDM a été repéré par les grands constructeurs automobiles et les fournisseurs qui ont initié un projet avec un objectif de fournir un standard commun fondant sur les principes de l'IDM. Il est appelé AUTOSAR (Automotive Open System Architecture) et actuellement, ce standard est la plus influente en termes de modélisation des systèmes automobiles. La chaîne de développement de la méthodologie AUTOSAR (voir Figure 2) s'étend à partir de la représentation de composants logiciels d'application à l'infrastructure d'exécution, y compris la description de la plate-forme matérielle. Un inconvénient de l'AUTOSAR est son manque de soutien pour la modélisation du niveau de la fonction. Par conséquent, il ya un intérêt dans la combinaison de ce standard avec le langage de modélisation EAST-ADL2 qui prend en charge la spécification fonctionnelle.

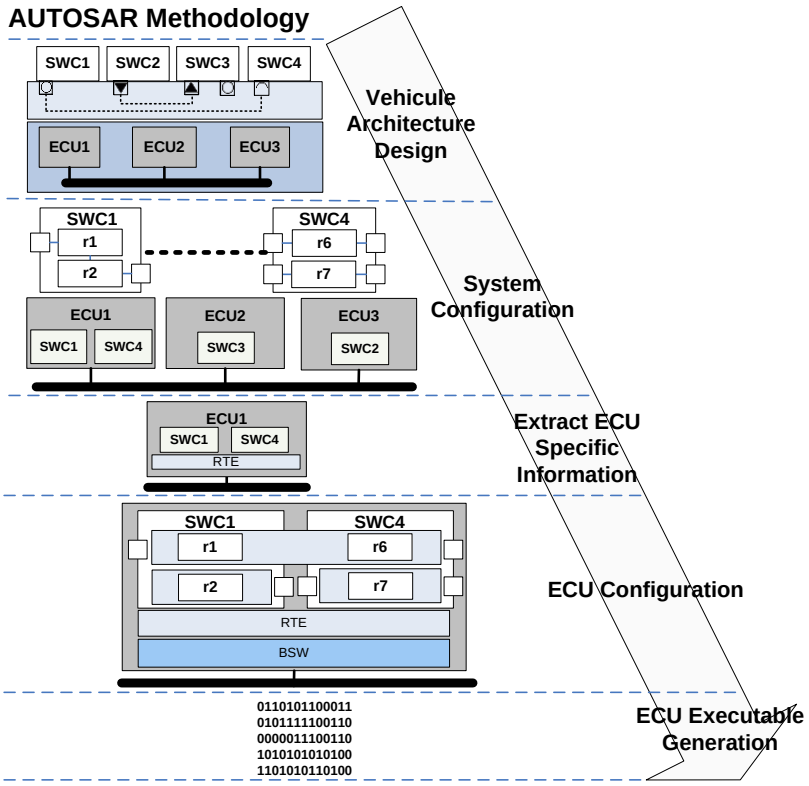

Figure 2. AUTOSAR Méthodologie

Le EAST-ADL2 et AUTOSAR imposent des règles méthodologiques pour la construction des modèles. Leur avantage est qu'ils fournissent un cadre commun pour la conception de systèmes électroniques automobile. Toutefois, aucune de ces définit comment effectuer certaines étapes de conception, par exemple, la façon de distribuer les composants logiciels sur les éléments matériels ou comment mapper des entités fonctionnelles sur des tâches OS (ang. Operating System). À cet égard, ces deux standards comptent entièrement sur une expérience de concepteur, augmentant ainsi le potentiels nombre de défauts de conception. En conséquence, il est essentiel de procéder à une analyse comme, l'analyse temporelle ou l'analyse de sécurité pour assurer que les décisions prises par le concepteur n'a pas conduit à des architectures irréalisables. Nous pouvons aller encore plus loin et utiliser des techniques pour la exploration de l'espace de conception (ang. DSE – Design Space Exploration). Leur emploi pourrait assurer la faisabilité, mais en plus permet d'optimiser les clés propriétés non-fonctionnelles.

2. *Énoncé du problème & motivation*

Comme indiqué dans le paragraphe précédent, vision claire et exhaustive sur une conception de système automobile, ainsi que son analyse / optimisation, sont les activités

nécessaires pour rester compétitif sur le marché de l'automobile. Cela nécessite des langages de modélisation, de méthodes d'analyse et des techniques pour permettre DSE. L'objectif général et initial de cette thèse est d'intégrer ces trois activités dans un cadre méthodologique, soutien de la conception des architectures automobiles et suivie par la méthodologie EAST-ADL2/AUTOSAR. Dans ce cadre, un ensemble de problèmes excité. Recherche de solutions appropriées est important de rendre possible l'intégration définitive de ces activités et la fourniture d'un flux continu guidé entre eux pour finalement livrer modèle d'implémentation optimisé d'un système automobile.

Ayant différents types de modèles, c.-à-d. le modèle d'architecture, modèle d'analyse et modèle d'optimisation, est l'étape principale vers la possibilité d'effectuer une synthèse optimisée du logiciel avec le matériel. La phase principale de la synthèse est appelé déploiement. Selon AUTOSAR, le déploiement concerne 1) **l'allocation** des composants logiciels sur ECU 2) **le partitionnement** des entités du comportement du composant (appelées runnable entities) sur des OS tâches et enfin 3) **l'ordonnancement** des OS tâches. Un point crucial pour cette étape est sa validité en fonction de ses propriétés temporelle. Depuis le raffinement du système (dont le déploiement est une partie intégrale) est fait top-down, la validité peut être assurée sous certaines hypothèses concernant des détails de niveau inférieur. Un exemple typique est l'hypothèse sur la connaissance des temps d'exécution pire cas (WCETs) des entités exécutables AUTOSAR. **Il est évident que l'hypothèse de la connaissance précise des WCETs de runnables avant l'implémentation du code est la plupart du temps irréaliste.** Dans de nombreux cas, l'implémentation de certaines runnables est réutilisée par les systèmes précédents, d'où leur WCET est connu. Cependant, ce n'est pas le cas quand les nouvelles runnables sont introduits. **Cela provoque le problème dans la tentative de remise des stratégies guidées pour la synthèse de l'architecture et, en général, la fourniture d'un flux top-down. Ce qui est encore plus important est que le déploiement défini, comme dans le AUTOSAR n'est pas soutenu dans la manière holistique par les techniques existantes. Bien que la quantité de travail qui existe semble être convaincant et représentatif, il y a un fossé. Techniques proposées soit représentent les OS tâches que les entités d'allocation ou résoudre le problème dans les étapes sans tenir compte d'un impact négatif qu'elle a sur une des résultats finales par rapport à l'approche holistique.**

EAST-ADL2 et les spécifications AUTOSAR offrent un large éventail de concepts qui sont nécessaires pour définir l'architecture complet du système. Les efforts récents pour étendre ces standards ont fourni les capacités à modéliser les informations nécessaires à

l'analyse temporelle. Le travail adéquate n'a pas été fait jusqu'à présent pour gérer les optimisations. **Bien que le domaine en temps réel et des systèmes distribués est riche en techniques d'optimisation, il n'y a pas de concepts de modélisation qui permettraient spécifiant une entrée nécessaire pour cette activité tels que les objectifs d'optimisation (temps des réponses, la consommation de mémoire, etc.).** En conséquence, la modélisation et l'analyse/l'optimisation ne sont pas bien intégrés. Cela a abouti à de nombreux outils soit pour la modélisation ou l'analyse et/ou l'optimisation.

3. Contributions

Afin de permettre de développement sans couture dans le cadre propose, ce travail propose un ensemble de solutions aux problèmes mentionnés ci-dessus:

1) Du côté des techniques de DES les principales contributions portent sur la définition de nouvelles techniques pour optimiser le déploiement. Les techniques proposées sont conformes à la définition de déploiement comme inclus dans le standard AUTOSAR. C'est-à-dire, ils considèrent les runnable entities que les unités d'allocation. Par conséquent, l'étape de partitionnement qui n'est pas considéré par les approches existantes est supportée par la technique défini dans ce travail. Techniques proposées sont basées sur quelques heuristiques, algorithme évolutionniste, diviser pour régner, amélioration itérative, c'est pourquoi ils sont capables de traiter de grandes architectures d'entrée. Cette caractéristique a été évaluée en effectuant plusieurs tests, atteignant 250 runnables. Aspect d'accompagnement qui a été évalué c'est la qualité des architectures déployées. Ceci a été réalisé en comparant les résultats à ceux obtenus avec les méthodes exactes ou des architectures pour lequel la configuration optimale de déploiement était connu a priori. Dans le AUTOSAR, des modèles de comportement pourraient correspondre soit à *data driven* ou *time driven* sémantique d'exécution. Cela nécessite de définir les différents types de stratégies d'optimisation. La différence se produit au nom de l'analyse d'ordonnançabilité qui diverse. En outre, les mesures d'optimisation tels que le métrique temporelle, le métrique du mémoire sont affectés dans une manière différente par les choix particuliers d'un déploiement. Ce qui caractérise aussi les techniques proposées est la prise en compte de multiples critères (par exemple, les réponses de bout-en-bout, les propriétaires temporelles, la consommation de mémoire) qui définit une bonne configuration de déploiement de l'architecture d'entrée. La Figure 3 montre un exemple de l'architecture logicielle d'entrée (partie supérieure) et sa spécification de déploiement.

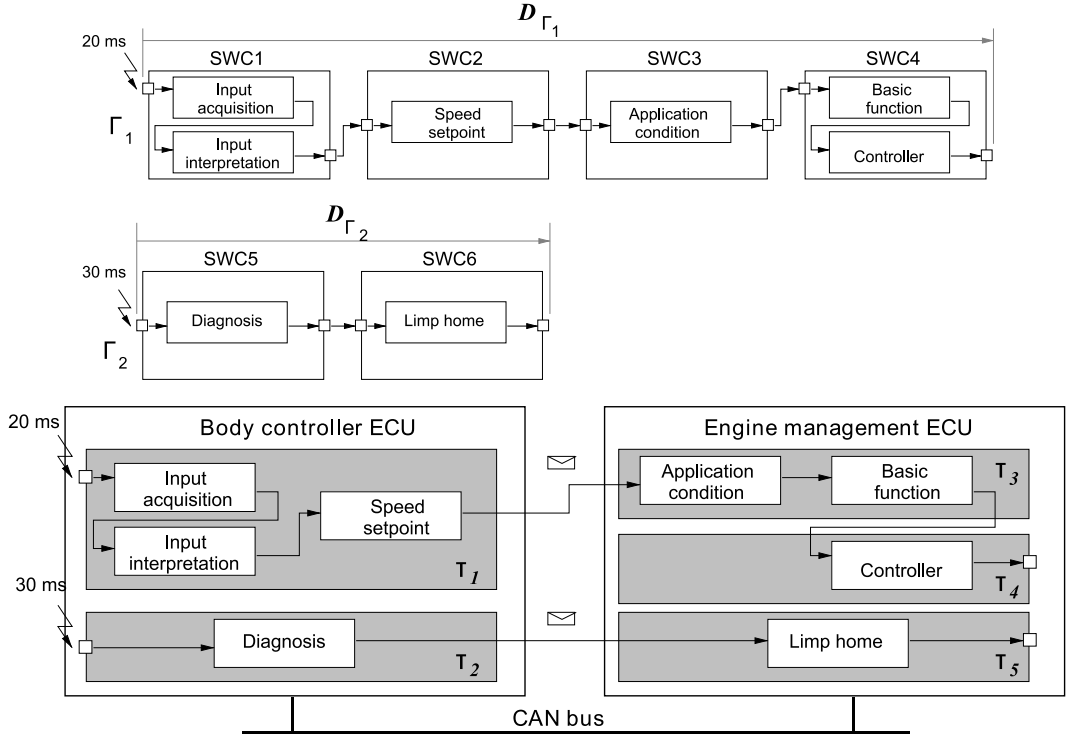

Figure 3. Exemple de l'architecture logicielle d'entrée (partie supérieure) et sa spécification de déploiement (partie basse)

.

2) Pour améliorer les résultats d'un déploiement, ce travail suggère un raffinement de la méthodologie EST-ADL2/AUTOSAR. Le but de le faire est de permettre de résoudre dans la manière holistique du problème de déploiement, qui ne peut être fait comme ça avec la définition actuelle de cette méthodologie. Le changement concerne la répartition des responsabilités entre les deux niveaux, le niveau fonctionnel couvert par EST-ADL2 et le niveau implémentation couverte par le AUTOSAR. L'activité *Design* qui se fait au niveau fonctionnel comprend l'étape d'allocation des fonctions atomiques aux ressources matérielles, c.-à-d. ECUs. Ceci détermine la répartition des runnable entities en raison de l'hypothèse dans laquelle les runnable entities sont transformés à partir des fonctions atomiques. C'est pourquoi le problème de déploiement ne peut pas être considéré dans la manière holistique au niveau de AUTOSAR parce que une dimension du problème, c.-à-d. l'allocation est déjà fixé. Par conséquent, ce travail préconise le changement dans lequel l'allocation est reportée jusqu'à le niveau d'implémentation. Évaluation de ce changement a été fait montrant une amélioration remarquable.

3) Pour effectuer un déploiement qui optimise les réponses de bout-en-bout, temps d'exécution des runnable entities sont nécessaires. Comme cette information peut-être manquant pour certains runnables, définition d'une nouvelle stratégie pour la configuration de

l'architecture est inévitable. Pour contourner le problème, certains travaux proposent d'ajouter à la méthodologie d'une activité spéciale - budgétisation de temps (ang. time budgeting). Au lieu d'estimer le pire des cas le temps d'exécution, l'intégrateur de système spécifie budgets dits temps, c'est à dire les contraintes de temps de réponse le pire des cas. Au lieu d'estimer WCETs, l'intégrateur de système spécifie budgets temps (ang. engtime budgets), c'est à dire des contraintes à des temps de réponse pire cas - WCRTs (ang. Worst Case Execution Times). Les budgets temps doivent être respectées par les fournisseurs livrant l'implémentation des composants. Le problème typique de cette approche est que le fournisseur livre l'implémentation d'un composant particulier, qui sera intégrée par l'intégrateur en tant que partie interactif du système, dans une étape ultérieure. Depuis le fournisseur valider le composant dans l'isolement, sans tenir compte d'éventuelles interférences d'autres composants, il sera incapable de calculer un temps de réponse pire cas correcte (WCRT). C'est-à-dire si le composant répond à la contrainte du budget temps l'intégrateur du système doit prendre soin d'éviter toute interférence possible avec d'autres composants. Ce n'est pas seulement une tâche difficile, mais qui provoque généralement surdimensionnement des ressources. Cette surdimensionnement des ressources se transforme en coûts insoutenables pour une production en série. Une solution alternative est celle dans laquelle les budgets temps représentent des contraintes de WCET de runnable entity à la place de sa WCRT. Ce travail propose une solution de budgétisation des WCETs. La plupart des travailles existants budget WCRT qui ne correspond pas bien à l'idée de « l'architecture intégré » soutenu par AUTOSAR. Un autre avantage de la technique proposée dans ce travail par rapport aux approches existantes est l'hypothèse que le déploiement n'est pas connu à l'avance. En conséquence, un objectif de la technique proposée est de trouver conjointement le déploiement et l'affectation optimale des budgets temps. La Figure 4 illustre un l'architecture logicielle d'entrée et l'architecture matérielle pour lesquelles le déploiement ainsi que les budgets de temps doit être spécifié. En fait, les budgets temps doivent être définies pour ces runnable entities pour lesquels les informations sur le WCET n'est pas présent. La Figure 5 présente le résultat de la technique proposée, c.-à-d. l'architecture déployée ainsi que les budgets temps.

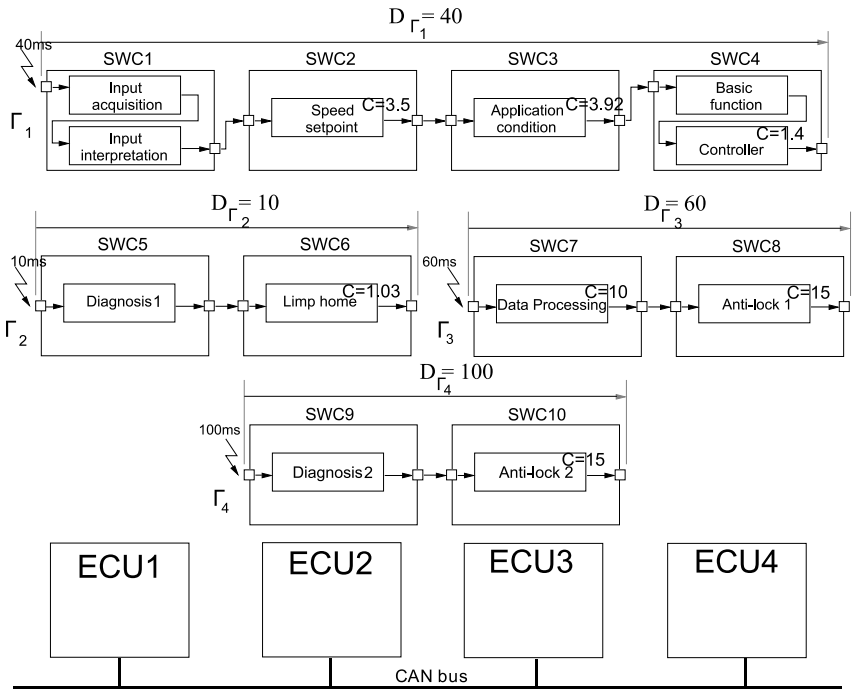

Figure 4. L'architecture Logicielle et Matérielle avec certains Runnables qui manquent WCETs

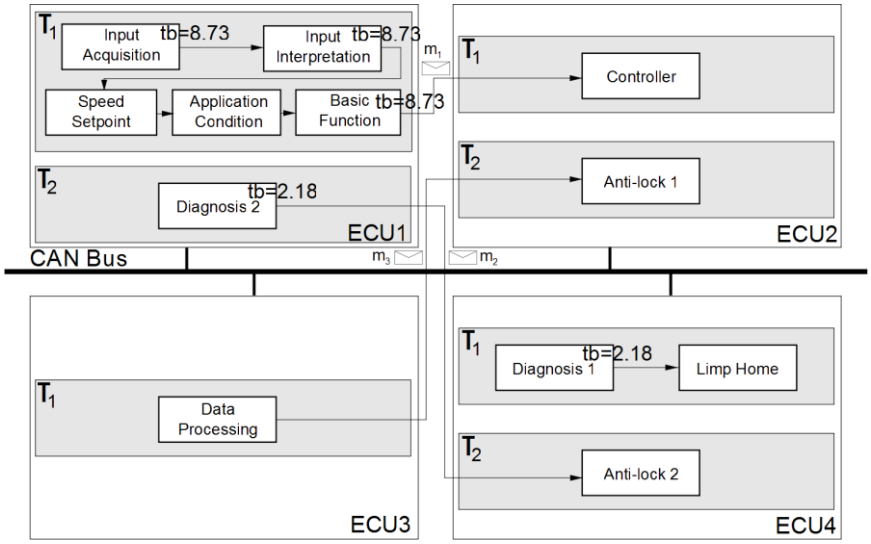

Figure 5. Résultant Déploiement avec la Spécification des Budgets Temps

4) En ce qui concerne la modélisation de la première contribution est une spécification de concepts essentiels pour construire un modèle d'optimisation et d'exécuter des techniques DES telles que celles définies dans ce travail, servant pour le déploiement ou budgétisation de temps. La Figure 6 représente une partie du profil UML définissant les principaux concepts permettant de construire contexte d'optimisation. Au-dessus du modèle d'optimisation, des techniques d'optimisation peuvent être exécutés.

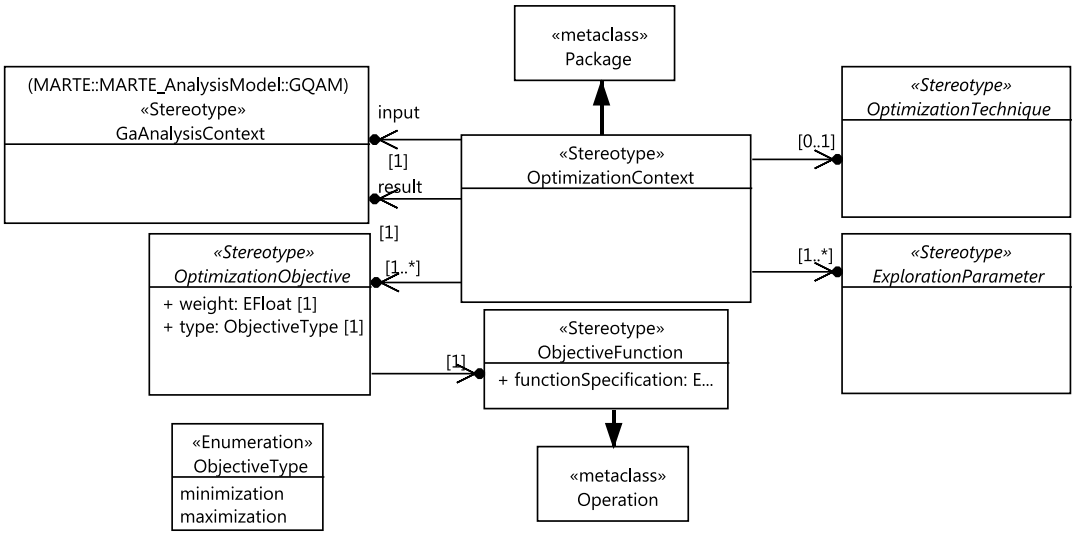

Figure 6. UML Profile pour le contexte d'optimisation

5) Les modèles d'optimisation ainsi que des modèles pour l'analyse et la spécification de l'architecture sont basés sur UML. L'utilisation de l'UML permet de faciliter l'intégration des différentes activités. Ceci est obtenu en majeure partie par l'ensemble des transformations qui automatisent des étapes importantes telles que la production du modèle AUTOSAR préliminaire a partir de modèle EAST-ADL2. En fait, la spécification d'architecture fonde sur les concepts de EAST-ADL2 et AUTOSAR et de les modéliser ce travail utilise un mécanisme de profil UML. Le profil UML complet pour l'EST-ADL2 déjà existe. Ce n'est pas le cas pour la AUTOSAR et donc ce travail définit un. Les modèles d'analyse sont établis avec le SysML et MARTE pour lesquels les profils UML ont été définis et standardisés par l'OMG (Object Management Group). Les concepts pour l'optimisation ne peuvent pas être exprimés ni avec le SysML ni MARTE ainsi que l'EST-ADL2 et AUTOSAR. En conséquence, pour eux, un modèle de domaine est formalisée et son profil UML est défini qui a déjà été mentionné dans le cadre de la contribution nr 4.

Tous ces modèles, modèles d'architecture, d'analyse et d'optimisation avec des algorithmes d'analyse et d'optimisation qui peut être exécuté sur eux ont été intégrés dans un cadre et structurés le long des couches d'abstraction et les points de vue. Le cadre lui-même (appelé AFfMAO – Architecture Framework for Modeling Analysis and Optimization) a été développé comme une instance d'un Cadre d'Architecture de l'Automobile (ang. Automotive Architecture Framework - AAF) définie dans le cadre de ce travail. AAF a été construit en suivant les principes du Cadre de l'Architecture (ang. Architecture Framework - AF) définie dans la norme ISO 42010. Cette relation est représentée sur la partie gauche de la Figure 7. En substance, le cadre d'architecture est un ensemble de conventions, principes et pratiques pour

la description des architectures dans un domaine et/ou communauté des parties prenantes. Par conséquent la spécification de l'AAF a été fait en définissant des points de vue de l'architecture avec leurs préoccupations, sortes de modèles et de règles de correspondance. Le côté droit de la Figure 7 présente perspective détaillée sur la AFfMAO. Informations pertinentes à partir de ce figure concerne les choix des techniques de modélisation, un ensemble de transformations, des algorithmes d'analyse et d'optimisation et plate-forme utilisée pour réaliser l'AAF comme AFfMAO.

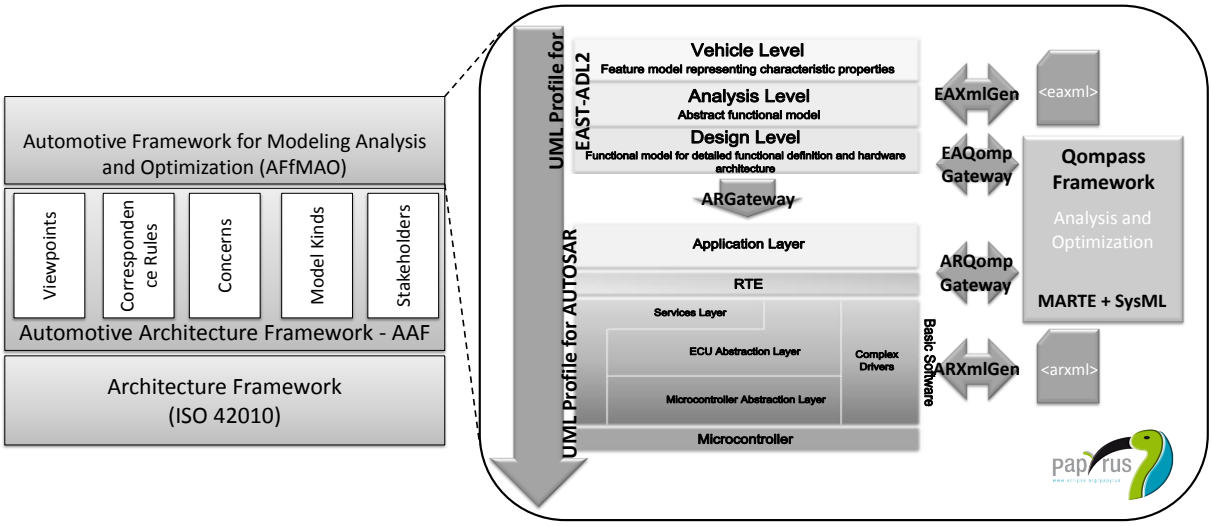

Figure 7. AFfMAO construit comme une instance de l'AAF

Contributions établies résoudre les problèmes cruciaux qui entravent la livraison d'un cadre pour une désign guidée des systèmes automobiles, alignés sur les principes de l'ingénierie dirigée par les modèles. Ils sont bénéfiques non seulement dans le contexte de ce cadre particulier, mais en général à ces constructeurs qui tente d'engager le standards EAST-ADL2 et AUTOSAR que la base de leurs systèmes.