

Modèles et transformations de modèles pour l'introduction de la consommation énergétique dans le standard AUTOSAR

Borjan Tchakaloff, Sébastien Saudrais, Jean-Philippe Babau

▶ **To cite this version:**

Borjan Tchakaloff, Sébastien Saudrais, Jean-Philippe Babau. Modèles et transformations de modèles pour l'introduction de la consommation énergétique dans le standard AUTOSAR. Conférence en IngénieriE du Logiciel (CIEL), Jun 2012, Rennes, France. papier n°20. hal-00734226

HAL Id: hal-00734226

https://hal.univ-brest.fr/hal-00734226

Submitted on 21 Sep 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Modèles et transformations de modèles pour l'introduction de la consommation énergétique dans le standard AUTOSAR*

Borjan Tchakaloff
ESTACA, Laval
Lab-STICC, UEB, UBO, Brest
borjan.tchakaloff@estaca.fr

Sébastien Saudrais
ESTACA, Laval
sebastien.saudrais@estaca.fr

Jean-Philippe Babau
Lab-STICC, UEB, UBO, Brest
jean-philippe.babau@univ-brest.fr

Résumé

Dans un véhicule automobile électrique, la connaissance et la maitrise des consommations énergétiques est primordiale pour une gestion optimale de l'autonomie du véhicule. La gestion logicielle des équipements s'appuie sur une architecture AUTOSAR à base de composants. AUTOSAR ne permet pas actuellement de modéliser les aspects énergétiques des divers équipements. Pour pallier ce manque, ce papier propose un premier modèle de consommations liées à l'utilisation des équipements automobiles. Ce modèle est utilisé pour générer — à partir d'un architecture fonctionnelle AUTOSAR — un modèle d'architecture AUTOSAR enrichi de politiques de contrôle de la consommation énergétique. Nous illustrons cette proposition avec une étude de cas (gestion du plafonnier d'un véhicule).

Mots-clefs : **systèmes embarqués**, **AUTOSAR**, **composants logiciels**, **transformation de modèles**, **gestion de la consommation énergétique**

Abstract

In an electric automotive vehicle, knowing and mastering the energy consumption of the different parts is essential to allow optimal management of the vehicle autonomy. Software handling of devices relies on an AUTOSAR component-based architecture. Though, AUTOSAR does not allow one to model the energy requirements of the vehicle equipments. The approach presented in this paper aims to offer a solution by proposing a first energy model of the various devices' consumptions. This model is used to generate — from a functional AUTOSAR architecture — an enhanced AUTOSAR architecture model with energy-consumption management policies. We illustrate this proposal with a case study (vehicle's dome light management).

Keywords : **embedded systems**, **AUTOSAR**, **software components**, **models transformation**, **energy consumption management**

1 Introduction

Les véhicules électriques sont désormais accessibles au grand public. Les véhicules de petit gabarit permettent d'assurer une autonomie d'une centaine de kilomètres, assurant les déplacements journaliers d'une famille. La plupart des fonctionnalités présentes dans les véhicules sont réalisées par fonctions logicielles, à l'exception de quelques-unes comme le freinage.

*Thèse financée et réalisée au sein de l'équipe Systèmes Embarqués de l'ESTACA.

La conception logicielle des différents systèmes embarqués présents au sein des véhicules s'appuie sur le standard ouvert AUTOSAR (*AUTomotive Open System ARchitecture*) [1] développé par un consortium de constructeurs, d'équipementiers automobiles et de fournisseurs d'outils logiciels. Ce standard permet de répondre aux manques d'interopérabilité et de réutilisation dont souffraient les systèmes logiciels automobiles grâce à une architecture à composants. La modélisation AUTOSAR concerne l'implémentation des calculateurs, et l'architecture sous la forme d'une vue générale fonctionnelle.

Le système de stockage énergétique — que nous simplifions par *batterie* [1] — est une ressource limitée. Lors de l'utilisation d'un véhicule électrique, le conducteur souhaite s'assurer que l'état de charge de la batterie soit suffisant pour le trajet qu'il souhaite effectuer. Ceci demande une prédiction de la consommation énergétique des différents organes du véhicule utilisés par les fonctions actives. Dans les véhicules présents sur le marché, la surveillance du niveau de la batterie est sommaire. Les fonctionnalités du véhicule sont bridées lorsque le niveau est trop faible mais l'objectif du trajet n'est pas analysé.

Afin d'améliorer les performances (temps d'utilisation du véhicule) et le confort (durée d'utilisation des services), nous proposons de modéliser les consommations énergétiques des équipements par le biais des composants qui y sont liés. Nous introduisons également dans ce modèle les politiques de gestion énergétique qui régulent les fonctionnalités liées aux équipements. Un deuxième modèle permet d'architecturer la solution logicielle de gestion énergétique, selon un schéma compatible avec le méta-modèle d'AUTOSAR. Notre approche permet finalement d'obtenir un modèle AUTOSAR enrichi des politiques de gestion énergétique.

2 Proposition

L'idée de l'approche est de modéliser les différentes consommations et politiques de gestion énergétique des équipements, puis par transformations successives de modèles, de raffiner le modèle fonctionnel initial afin d'obtenir un modèle enrichi des politiques de gestion de l'énergie. Le résultat reste compatible avec le méta-modèle d'AUTOSAR afin d'assurer une compatibilité avec les outils existants.

2.1 Architecture

La solution retenue privilégie la séparation des préoccupations via des modèles dédiés et l'automatisation du processus par des transformations d'initialisation des modèles et de génération de modèles. La figure 1 illustre les liens entre les modèles et les transformations utilisés.

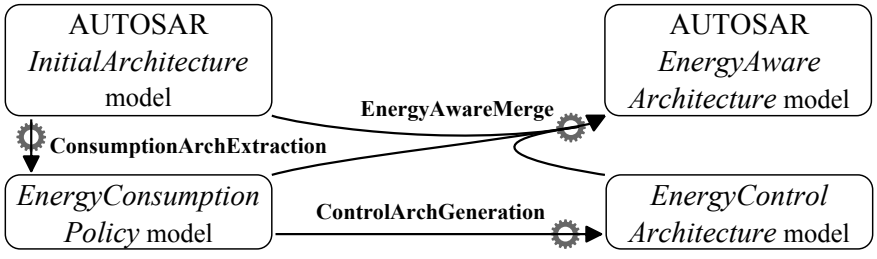


FIGURE 1 – Modèles et transformations de modèles composant notre proposition

1. Le stockage énergétique est généralement réalisé par une batterie (ou un pack de batteries) ou par une hybridation batterie–super-condensateurs.

Dans un premier temps, les composants logiciels liés aux équipements consommateurs d'énergie sont extraits du modèle fonctionnel AUTOSAR *InitialArchitecture* à l'aide d'une première transformation *ConsArchExtraction*.

Le modèle de consommation *EnergyConsumptionPolicy* est initialisé via cette première transformation : mise en place de la structure du modèle. Ce modèle est composé de deux parties. Dans une première partie, les consommations des divers équipements sont modélisés. La deuxième partie modélise les règles de gestion logicielle énergétique des équipements. Pour les deux parties, une librairie de modèles types est proposée afin d'aider à leur mise au point.

Une deuxième transformation *CtrlArchGeneration* permet, à partir du modèle *EnergyConsumptionPolicy*, de créer un modèle architectural énergétique *EnergyControlArchitecture*. Ce modèle regroupe l'ensemble des composants dédiés à la gestion énergétique, modèle compatible avec AUTOSAR.

Enfin, la fusion *EnergyAwareMerge* des modèles *InitialArchitecture*, *EnergyConsumptionPolicy* et *EnergyControlArchitecture* permet de générer le modèle AUTOSAR enrichi *EnergyAwareArchitecture*. La solution reste compatible avec le standard AUTOSAR et intègre de nouveaux composants implémentant les règles spécifiées dans l'*EnergyConsumptionPolicy*.

2.2 Architecture détaillée

Nous détaillons maintenant les différents modèles utilisés dans notre approche. Ils sont illustrés par un exemple de modélisation de plafonnier, une fonctionnalité non-critique gérable de manière automatique.

2.2.1 Le modèle AUTOSAR *InitialArchitecture*

Le standard AUTOSAR est utilisé pour modéliser les fonctionnalités d'un système automobile. La diversité des fonctionnalités rend le méta-modèle AUTOSAR riche et complexe. Nous décrivons ici les concepts AUTOSAR utilisés dans notre approche pour l'optimisation d'une ou plusieurs fonctionnalités sur une seule unité de traitement, appelée *ECU* (Electronic Control Unit). Un modèle AUTOSAR permet la description et la configuration d'une *ECU*. Une fonctionnalité est découpée en composants, nous en utilisons deux types : les composants *Sensor*s–*Actuator*s représentant les entrées–sorties de manière abstraite et les composants *Application*s encapsulant les fonctions applicatives. Les composants exportent leurs interfaces via des ports de données. Les connexions entre composants se font sur des ports compatibles. Il est à noter qu'il existe un composant spécial, l'*IOHardwareAbstraction*, auquel sont connectés tous les *Sensor*s–*Actuator*s. Il permet d'abstraire les entrées–sorties matérielles présentes sur l'*ECU*.

En plus des concepts présentés, nous définissons quelques contraintes sur les modèles d'entrée AUTOSAR. Notre approche requiert que certaines données de contexte soient disponibles comme le niveau d'énergie restante de la batterie, appelé *BatteryLevel*. Les composants *Sensor*s–*Actuator*s ont un port de contrôle leur permettant d'influer sur le matériel qu'il représente. Ce sont par exemple les différentes positions d'un commutateur physique. Les composants *Application*s connectés à ces ports sont appelés par la suite *Controller*s.

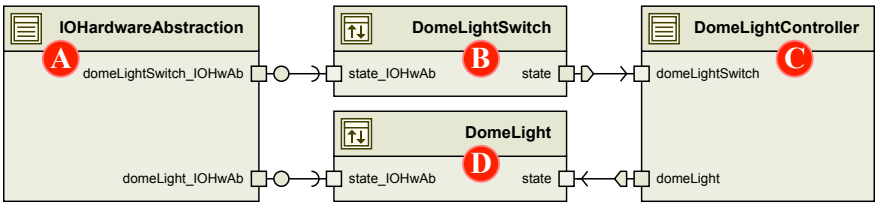


FIGURE 2 – Modélisation AUTOSAR du système de plafonnier

La figure 2 illustre les différents types de composants et leur connexions. Nous y trouvons les *Sensors–Actuators* représentant l'interrupteur du plafonnier (**B**) et le plafonnier (**D**). Ils sont connectés à l'*IOHardwareAbstraction* (**A**). Le *Controller* (**C**) connait l'état de l'interrupteur et modifie l'état du plafonnier.

2.2.2 Le modèle énergétique *EnergyConsumptionPolicy*

Le modèle énergétique est composé de deux parties : la consommation des composants et les règles d'optimisation. Ce modèle est obtenu par l'extraction des composants du modèle AUTOSAR liés aux consommations énergétiques : *Sensors–Actuators* et *Controllers*. Cette extraction est réalisée par la transformation de modèles *ConsArchExtraction*.

Require: states of DomeLight **in** *Watt*
1: **states of** DomeLight
2: **state** OFF → 0
3: **state** ON → 10
4: **end-states**

FIGURE 3 – Exemple de consommations

Require: BatteryLevel **in** *Percent*
1: **when (** BatteryLevel = 50 **or** BatteryLevel = 40 **or** BatteryLevel = 30 **) do**
2: *InhibitTheMostHungryEnergyConsumer*
3: **done**
4: **when (** BatteryLevel = 25 **) do**
5: *InhibitComponent* DomeLight
6: **done**

FIGURE 4 – Exemple de règles

La première partie du modèle (Figure 3) permet de représenter les niveaux de consommation du matériel. Ces niveaux sont liés aux composants représentants les équipement, les *Sensors–Actuators*. Nous illustrons cette partie par des consommations simples (puissances constantes dans le temps) mais les profils de consommations (puissances fonction du temps) sont également modélisables. Les puissances modélisées permettent de connaitre l'énergie consommée grâce à la formule $énergie = puissance * temps$.

La seconde partie (illustrée par la Figure 4) permet au concepteur de choisir les règles de gestion énergétique à utiliser parmi la librairie fournie. Les règles sont paramétrables et sont basées sur la logique d'événement/action. Plusieurs événements différents peuvent déclencher une même règle (ligne 1) qui entraîne un ensemble d'actions. Les actions ciblent des composants selon leurs propriétés (ligne 2) ou de manière nommée (ligne 5).

2.2.3 Le modèle architectural *EnergyControlArchitecture*

Le second modèle utilisé décrit l'architecture des composants réalisant les politiques de gestion d'énergie. Cette architecture est générée par la transformation *CtrlArchGeneration* à partir des politiques de gestion définies dans le modèle précédent.

L'architecture proposée s'appuie sur trois composants types. L'*EnergyManager* supervise la consommation énergétique de l'*ECU* sur laquelle il se situe à l'aide des politiques de gestion énergétique. L'*EnergyContextObserver* permet de récupérer les informations contextuelles (dont la valeur de *BatteryLevel*). Les *EnergyBroker*s sont gérés par l'*EnergyManager* et exécutent ses décisions au niveau des composants liés aux équipements consommateurs. Ils remplacent les connexions existantes entre les *Sensors*–*Actuator*s et leurs *Controller*s. Dans notre exemple, les *EnergyBroker*s sont des inhibiteurs : ils laissent passer les requêtes des *Controller*s ou non, en fonction de ce que leur dicte l'*EnergyManager*.

2.2.4 Le modèle AUTOSAR enrichi *EnergyAwareArchitecture*

Pour créer le modèle AUTOSAR final enrichi, on fusionne le modèle initial, le modèle énergétique et le modèle architectural par la transformation *EnergyAwareMerge*.

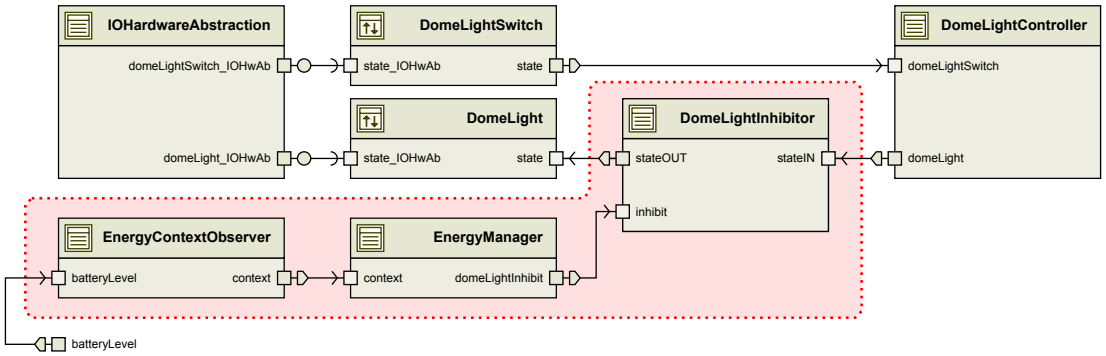


FIGURE 5 – Modélisation AUTOSAR enrichie

Nous obtenons finalement un modèle AUTOSAR enrichi des données de consommations encapsulées dans un *EnergyManager*. Cette transformation intègre les composants créés dans le modèle architectural en modifiant les connexions existantes et génère leur implémentations à partir du modèle énergétique. L'*EnergyContextObserver* est uniquement prototypé.

La figure 5 montre l'intégration des nouveaux composants (à l'intérieur du cadre pointillé) et comment les connexions entre les *Sensors*–*Actuator*s et leurs *Controller*s sont remplacées par les *EnergyBroker*s — ici un inhibiteur qui en est une spécialisation — grâce à la méthode présentée dans [6].

3 Travaux connexes

Becker et al. présentent une approche de reconfiguration en ligne dans [2]. L'approche proposée repose sur la modélisation de configurations, état du système où certains composants sont actifs, via un outil dédié qui permet de mettre à jour le modèle AUTOSAR. La méthode proposée impose une définition de toutes les configurations à la modélisation du système : la reconfiguration est donc définie de manière statique.

Chauvel et al. [3] utilisent des règles basées sur la logique floue pour reconfigurer dynamiquement des propriétés de composants et leur architecture. Dans notre cas, l'utilisation de règles utilisant la logique floue permettrait aux concepteurs de règles une approche naturelle et souple en utilisant des termes usuels tels que : « niveau de batterie faible », « niveau de batterie critique », « l'utilité d'arrêter ce composant est importante ».

Il existe de nombreux styles architecturaux ([5] [7] [8]) pour gérer la qualité de service (QdS) au sein des architectures à composants. On retrouve classiquement certains composants types tels les *Manager*s pour définir les politiques de gestion de la QdS, les *Broker*s pour gérer les composants et les *Observer*s pour surveiller le contexte et l'exécution.

4 Conclusion

Dans ce papier, nous avons présenté une modélisation des consommations énergétiques des équipements pour AUTOSAR afin d'améliorer la gestion de l'énergie à bord d'un véhicule. Nous introduisons les consommations et les politiques de gestion dans un premier modèle *EnergyConsumptionPolicy* à partir d'un modèle fonctionnel AUTOSAR. Un second modèle *EnergyControlArchitecture* définit les composants logiciels dédiés à la gestion énergétique. Nous obtenons finalement un modèle AUTOSAR enrichi des politiques de gestion énergétique en fusionnant les précédents modèles.

Les travaux futurs porteront sur l'enrichissement de la librairie des politiques de gestion, définies actuellement de manière locale à une *ECU*. L'introduction de politiques globales permettrait d'harmoniser la gestion de l'énergie sur l'ensemble du véhicule. L'introduction de nouveaux composants dans un système fonctionnel augmente les taux d'occupation *CPU* et réseau et peut entraîner la violation des contraintes constructeurs. Une optimisation de l'allocation des composants du modèle enrichi est réalisable avec des techniques de *Design Space Exploration* [4] afin de respecter ces contraintes.

La validation de notre approche se fera par expérimentation sur cible, sur la plateforme NEC V850 compatible AUTOSAR, avec un cas d'utilisation industriel.

Références

[1] AUTOSAR. `http://autosar.org`.

[2] Basil Becker, H. Giese, S. Neumann, M. Schenck, and A. Treffer. Model-Based extension of AUTOSAR for architectural online reconfiguration. In *Proceedings of the 2nd International Workshop on Model Based Architecting and Construction of Embedded Systems (ACES-MB 2009)*, volume 507 of *CEUR Workshop Proceedings*, pages 123–137. CEUR-WS.org, 2009.

[3] Franck Chauvel, O. Barais, I. Borne, and J.M. Jezequel. Composition of qualitative adaptation policies. In *Proceedings of the 2008 23rd IEEE/ACM International Conference on Automated Software Engineering*, pages 455–458, 2008.

[4] Ahmed Daghsen, K. Chaaban, and S. Saudrais. Software function allocation and configuration of an autosar-compliant system. In *SAE 2012 World Congress & Exhibition*, 2012.

[5] Prakash Manghwani, J. Loyall, P. Sharma, M. Gillen, and J. Ye. End-to-end quality of service management for distributed real-time embedded applications. In *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05)-Workshop 2-Volume 03*, page 138. IEEE Computer Society, 2005.

[6] Sébastien Saudrais and K. Chaaban. Automatic relocation of autosar components among several ecus. In *CBSE '11 : Proceedings of the 14th international ACM Sigsoft symposium on Component based software engineering*, 2011.

[7] Michal Sojka and Z. Hanzálek. Modular architecture for real-time contract-based framework. In *Industrial Embedded Systems, 2009. SIES '09. IEEE International Symposium on*, pages 66 –69, 2009.

[8] Jean-Charles Tournier, J.P. Babau, and V. Olive. Qinna, a component-based QoS architecture. *Component-Based Software Engineering*, pages 371–387, 2005.