Model transformation and scheduling analysis of an AUTOSAR system

Ahmed Daghsen, Khaled Chaaban, Sébastien Saudrais
ESTACA campus ouest
Embedded systems laboratory
Laval, 53000, France
ahmed.daghsen@estaca.fr

Mohamed Shawky
Université de Technologie de Compiègne
Heudiasyc laboratory
Compiègne, 60200, France
mohamed.shawky@hds.utc.fr

***Abstract*— AUTOSAR standard provides a common framework for software development in the automotive domain. It enables to manage the growing of the automotive architecture complexity by facilitating the integration and reuse of software components. However, additional work is needed to enable scheduling analysis and to handle with more timing properties in the system. In this paper, we propose an approach to enable a model transformation of the AUTOSAR timing model to a classical scheduling one. This allows to apply directly fundamentals scheduling theories for timing analysis. Then, we apply our approach through a steering-by-wire case study. Finally, we analyze the results given by the holistic algorithm and those given by a compositional one.**

I. INTRODUCTION

Many automotive applications are considered as time-critical or at least time-dependent. Thus, precise timing and prioritization of functions are essential for both safety and comfort of in-vehicle applications.The AUTomotive Open System ARchitecture (AUTOSAR) standard [1] is introduced to define a standard layered software architecture and interfaces. Many improvements and extensions to the current AUTOSAR system model have been developed recently to handle all timing-related information during the development process. Thus, complexity and development cost cycle are reduced significantly while reliability is improved. AUTOSAR allows an easy integration of timing information, however, few works use these timing properties and constraints to make a global timing analysis of the system. The local timing analysis addresses tasks scheduling regarding an Electronic Control Unit (ECU), and global scheduling considers the global distributed system where communication bus and gateways must be analyzed together with ECUs tasks. We can distinguish between local and global timing analysis. Where local timing analysis addresses tasks scheduling regarding a processor or an ECU, global scheduling considers the global distributed system where communication bus and gateways must be analyzed together with ECUs tasks. There are several problems related to such distributed system that must be addressed, such as task synchronizations and communication dependencies between processes. In this paper we propose an approach to enable a transformation of AUTOSAR timing properties and constraints into a complete scheduling model. By using this model, we can apply directly existing scheduling theories to the AUTOSAR application. As shown in figure 1 the approach consists of two main steps. The first step consists in applying a model transformation to convert AUTOSAR timing model to a scheduling model. The second step permits to apply the scheduling techniques for a global timing analysis of the AUTOSAR system. We show that this analysis allows to take into account further timing properties like task synchronizations, process communications modeled as offsets, jitter, constrained deadlines, process preemption and blocking overheads. The proposed method is applied to a steering-by-wire case study and we analyze scheduling results given by both a holistic and compositional scheduling approaches.

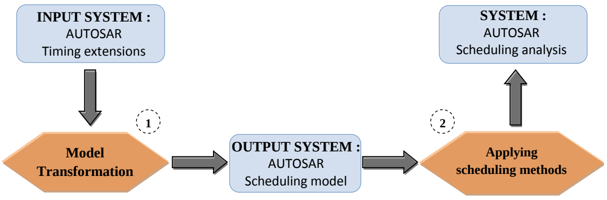

Fig. 1. AUTOSAR model transformation

II. AUTOSAR METHODOLOGY

In this section, we present a brief description of AUTOSAR methodology. Figure 2 describes the development process structure of an AUTOSAR software.

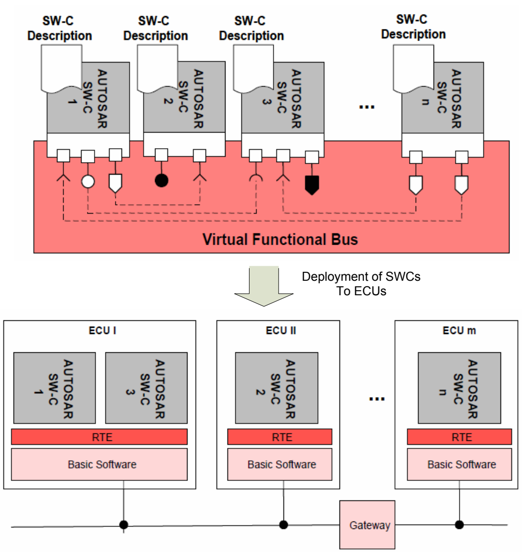

Fig. 2. AUTOSAR methodology

The first step consists on the definition of software components (SWCs) constituting the user software applications. SWCs communicate using ports through their interfaces. A SWC may be one of the three types: sensor/actuator, application or calibration type. Each SWC contains runnable entities which represents the C code that will be executed on the ECU. A runnable is triggered using an event which may be of timing or data type. In second step, at the Virtual Functional Bus (VFB) level, SWCs are defined without consideration of the underlying hardware on which these SWCs will run on later. So, two software components might run on the same ECU or on different ECUs and this is completely transparent to software developers. The communication between the components is then either an intra-ECU communication or an inter-ECU communication and is routed via the VFB bus which allows a virtual integration of the system independently of underlying software and hardware. Next, the mapping of the SWCs to available ECUs is performed. This phase requires some information about system and ECU constraints such as the input/output hardware connection. Finally, we can proceed by the development and integration of each ECU. The software architecture of an ECU is composed of three main layers: the SWCs, the Run Time Environment (RTE) and then the Basic Software (BSW) layer. The SWCs contain the application's functional code. RTE represents an instance of the VFB bus per ECU. It provides standardized interfaces to communicate with the BSW layer and to communicate between SWCs themselves. Data exchange between SWCs themselves and between SWCs and the underlying BSW layer is performed exclusively via RTE. Depending on SWCs locations, data exchange is performed either directly via a shared memory or by sending messages via a network bus. BSW layer makes the link between RTE layer and all hardware features of the ECU.

III. SYSTEM SCHEDULING ANALYSIS

In this section, we present related works dealing with timing analysis in AUTOSAR system and the existing formal approaches for system level performance analysis.

a) Scheduling in AUTOSAR: few works are dealing with the exploitation of AUTOSAR timing extensions for timing analysis. In the scope of TIMMO project, [6] gives a general framework for the relation between AUTOSAR concepts and timing constraints. They have proposed also an extension of AUTOSAR standard towards the possibility to specify the system's timing constraints. Thus, a scheduling analysis of an AUTOSAR application can be performed at the low-level. But the resulting task timing reveals hardly any direct timing-relation with high-level software components to which timing information shall finally be attached.

b) Compositional & Holistic scheduling: compositional performance analysis enables performance analysis for complex heterogeneous embedded architectures and supports subsystem integration [4]. This approach consists of integrating either offline or online scheduling analysis techniques into a system-level analysis. On the other hand, holistic analysis [8] introduced by Tindel, refers to a consistent end-to-end response time analysis approach for multi-processor real-time systems, where processors communicate over a bus and offline scheduling methods could be applied for timing analysis. The holistic technique also captures the timing using system-level equations. However, flexibility, subsystem integration and scalability are major weaknesses of holistic techniques. Recent works of Turja and Nolin [5] presented a method for calculating tighter (i.e. lower) response-times. Their method, under certain conditions, calculates the exact worst-case response time with offset. In practice the holistic approach is used in system configurations having low dependency complexity such as deterministic TDMA network. [2]

IV. TRANSFORMATION OF THE AUTOSAR MODEL

Our approach aims to perform a transformation of AUTOSAR timing properties and constraints into a complete scheduling model. By using this model, we can apply existing scheduling theories to the AUTOSAR application.

A. *AUTOSAR input model*

We consider an AUTOSAR system model as an input of the transformation process. The main timing related concepts of this model (AUTOSAR release 4.0) are:

- SWC: encapsulates a part of the functionality of the application.
- Event timing chain: Temporal correlation between two observable events.
- Period: Time interval between two consecutive event occurrences.
- Jitter: The maximum variation of timing event period
- Latency: The time duration between the occurrence of the stimulus and the occurrence of the response.
- Runnable: Is a part of an atomic software component which can be executed and scheduled independently.

In this model, a task is called an end-to-end chain which consists of a set of subchains and has an end-to-end deadline. Each subchain is assigned a proper priority and its worst-case response time can be bounded.

B. *Model transformation*

Table I illustrates some relationship between scheduling system model and the AUTOSAR timing concepts one. The output model consists of an end-to-end system model which is used as the basis of this work.

AUTOSAR model	**System model**
Subchain	Subtask
ECU	Processor
Communication bus	Link processor
Latency	Release time
Runnable	Subtask

TABLE I
RELATIONSHIP BETWEEN SCHEDULING SYSTEM MODEL AND AUTOSAR 4.0 CONCEPTS

In real-time system, an end-to-end system consists of more than one processor and a set of end-to-end tasks. In our model, the workload on a multiprocessor P_i system consists of a set T_i of end-bend tasks, each of which is a periodic task with period p_i, phase f_i, execution time t_i, and relative deadline D_i. In this paper, we assume that the relative deadline of a task is less than or equal to its period, i.e., $D_i \leq p_i$. The release time of the first instance of $T_{i,1}$ is the phase f_i of task T_i. An instance of $T_{i,j}$ cannot start to execute before the complete execution of $T_{i,j-1}$. A task T_i is a chain of subtasks $T_{i,j}$. Each subtask $T_{i,j}$ is one continuous execution thread of T_i on one processor and has a maximum execution time $Y_{i,j}$ and a fixed priority $prio_{i,j}$. Subtasks are statically assigned to processors. Subtask $T_{i,j}$ is a predecessor (successor) of subtask $T_{i,k}$ if $j < k$ $(j > k)$, and $T_{i,j}$ is the immediate predecessor (successor) of $T_{i,k}$ if they are also adjacent $(\mid j-k \mid = 1)$. The system model imposes strong restrictions on tasks properties. We consider both preemptive and non-preemptive tasks and subtasks. We also assume a common time base for all processors and we consider the jitter and the offset of periodic tasks.

V. CASE STUDY

The main objective of this section is to apply both the holistic and compositional analysis to a steering-by-wire case study. We begin by presenting the steering-by-wire system, which is developed in our laboratory. Then, we apply our approach using a holistic scheduling algorithm: Per Task Time Demand function (PTTDF). Finally, we simulate the system using a compositional scheduling tool and we analyse the results of each scheduling approach.

A. *Steering-by-wire system*

As depicted in Figure 3, a basic steering-by-wire system is composed of three main blocks: the hand wheel (i.e. steering), controllers and the road wheels. When the driver operates the hand wheel to turn the vehicle, a steering angle signal will be sent to the controller. Two kinds of sensors are necessary to acquire the steer angle and the torque applied by the driver. The controllers will process all acquiring signals and also perform

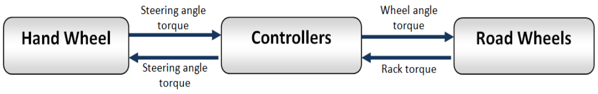

Fig. 3. Steering-by-wire system

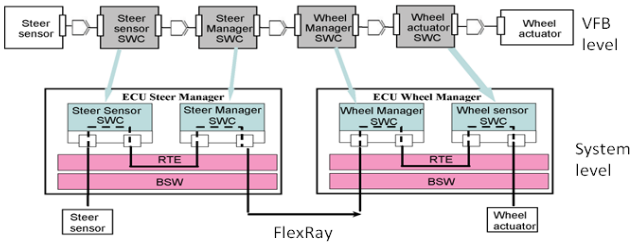

Fig. 4. Implementation of the rack torque function in AUTOSAR

some control functions associated with the vehicle's steering function and output an actuator angle for the road wheels that in turn will turn the wheels through an actuator. The feedback signals (actuator feedback and wheel feedback) involve some kind of force or torque sensors and are necessary so that the driver get the feeling of turning a traditional steering wheel and feel the effect of turning the wheels on a certain type of road. The steering-by-wire system may be composed of two main functions: the feedback torque function and the rack torque function. The feedback torque function permits to compute the feedback force applied to the steering wheel, so that the driver feels the effect of tuning the wheels on a certain type of road. The rack torque function is the main system function that permits to control the front axle actuator. The distributed steering-by-wire architecture involves several components: ECUs, Flexray bus for the communication lines and appropriate sensors and actuators. Flexray technology provides higher bandwidth, fast communication, fault tolerant and deterministic latency, enabling the development of innovative automotive system. Figure 4 illustrates the implementation of the rack torque function according to AUTOSAR approach. At the VFB level, the signal path involves four components. The ”Steer Sensor” component acquires the sensor physical data and passes it to the application software component ”Steer Manager”. Afterwards the signal is sent to the application software component ”Wheel Manager” for order computation until it is finally send to the actuator via the ”Wheel Actuator” component. At the system level, we map SWCs to available ECUs and then we configure RTE and BSW modules. In our case study we have only two ECUs: the steer ECU and the wheel ECU.

B. Applying scheduling algorithm to the AUTOSAR transformed model

We consider the system model obtained after the transformation process. Note that in AUTOSAR, and end-to-end task passes by three stages: from hardware to software represented by the transformation of data from the physical sensor to the sensor SWC (e.g. steer sensor or wheel sensor SWC), the second stage is all the actions that pass between the sensor SWC till the software control represented by the actuator SWC. The last stage is the interface that is done between the actuator SWC and the physical actuator (as shown in Figure 5). Each function, feedback torque function and rack torque function is represented as an end-to-end task. Then, the steering-by-wire system is represented by two end-to-end tasks. Let's note T_1 as the rack torque end-to-end task and T_2 as the feedback torque end-to-end task:
T_1: has 17 subtasks: $T_{1,1}$, $T_{1,2}$, $T_{1,3}$, $T_{1,17}$.
T_2: has 17 subtasks: $T_{2,1}$, $T_{2,2}$, $T_{2,3}$, $T_{2,17}$.
Each end-to-end timing chain segment in AUTOSAR model corresponds to a subtask. Subtasks of each function are executed on a specific ECU. As noted above, we have two processors: P1 at the steer side and P2 at the wheel side. The communication bus represents a link processor P3.

We remind that each process can be specified as a constrained deadline periodic task T=(O; J; p; t; D), where O is offset, J is jitter, p is period, t is the execution time, and $D(\leq T)$ is deadline. Priorities are fixed with respect of precedence constraint.

After having established the AUTOSAR scheduling model, we can now apply the PTTDF holistic algorithm. This algorithm presented in [7] allows to compute the tighter upper bounds of the response times of the end-to-end tasks in static systems. After computing the different equations of the algorithm, we obtain:

- The upper bound of task T_1: $C_1 = 133 > 50$ then T_1 is not schedulable.
- The upper bound of task T_2: $C_2 = 203.91 >> 50$ then T_2 is not schedulable.

Finally, according to the proposed algorithm and the obtained results the system is not schedulable.

C. Compositional analysis

In order to make a compositional analysis of the steering by wire system we use the SymTA/S tool [3]. It is a formal system-level performance and timing analysis tool of distributed systems. The approach is based on the compositional analysis. It permits to couple local scheduling analysis algorithms using event streams. Event streams describe the possible input/output timing of tasks.

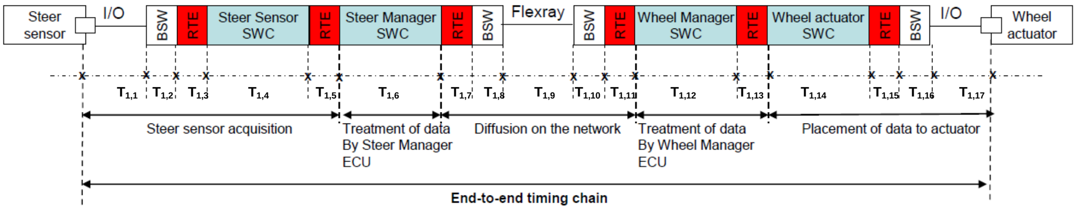

Fig. 5. End-to-end timing representation of AUTOSAR methodology

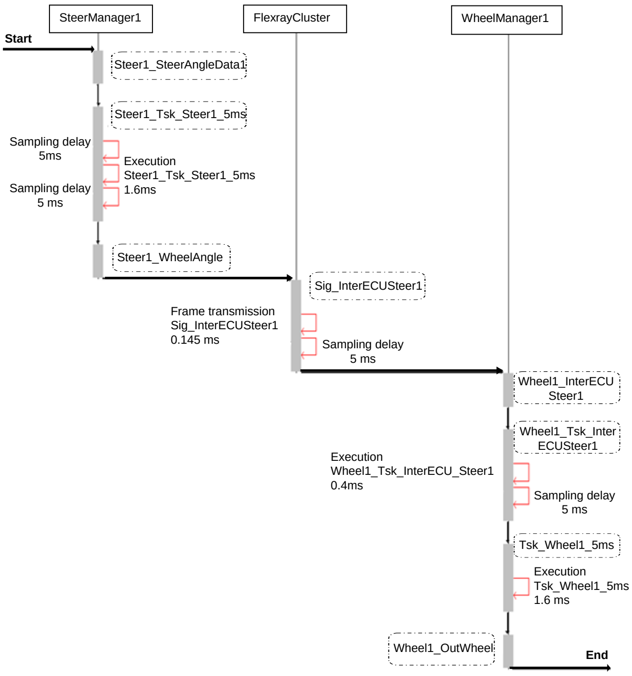

Fig. 6. End to end path of the rack torque function

By using SymTA/S analysis tool, we can calculate the local best-case and worst-case response times for tasks and frames, end-to-end best-case and worst-case response times for critical paths, deadline violations, utilization of resources, load contribution of individual tasks and individual frames.

Figure 6 illustrates the path of the rack torque function.

SymTA/S was used to model the steering-by-wire system with the timing properties like the task execution time, task periodicity, task execution type (preemptive, non-preemptive etc) etc. These timing models could define the system on design level and implementation level. The feasibility of the system was found out by using timing analysis feature available in the tool. Figure 7 shows the response time of the end-to-end path taking into consideration the synchronization between calculators, the offset and the jitter constraints.

D. Results and discussion

It has been shown that the holistic approach leads to pessimistic bounds. Since the used algorithm does not take into account the precedence constraints of subtasks when computing PTTD function which is the sum of all subtasks' execution time. This function assumes that the subtasks of each end-to-end task are independent, but the actual time demand may be less than the sum because of the precedence constraints among subtasks. Then the PTTD function can be considered as the maximum of the sum of all subtasks. Moreover, by considering the jitter and offset values, the obtained bound are higher. The holistic approach is more accurate by applying analytic equations. This technique is more suitable for system configurations with simplified equations such as deterministic Flexray and TDMA networks. The compositional analysis is a good candidate for more complicated heterogeneous system. The compositional model are well structured and uses event stream representation to allow component wise local analysis and also facilitates the subsystem integration.

VI. CONCLUSION

Performance analysis and timing requirements in AUTOSAR have received a wide attention recently. The main goal is to perform an early verification and analysis of the system performance at the design level and before implementation. There are few formal approaches to heterogeneous systems. In this paper, we have proposed an approach to transform an AUTOSAR timing model to a scheduling model. By this transformation, we can apply fundamentals scheduling techniques to the AUTOSAR system. Although, the proposed transformation is sufficiently generic to be integrated to other AUTOSAR architecture and communication paradigms like CAN or LIN, our approach makes several simplifications on the system model. However, a more accurate system model must be considered in order to take into consideration the architecture complexity, tasks interdependency and the low level layer interaction. Also, as a future work, we plan

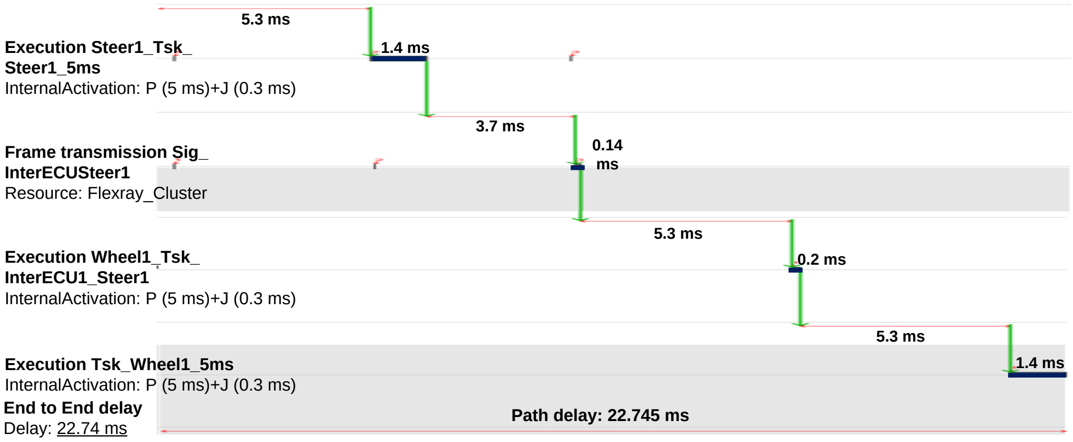

Fig. 7. Process with synchronization, offset = 0.2 and jitter = 0.3

to extend our system model by taking into consideration CPU resource sharing such as shared memory, multi-cores architecture and hardware constraints such as pipelining and caching.

REFERENCES

[1] Simon Fürst et al. Autosar - a worldwide standard is on the road. In *14th International VDI Congress Electronic Systems for Vehicles, Baden-Baden*, 2009.

[2] Rajesh K. Gupta and Giovanni De Micheli. *System-level Synthesis using Re-programmable Components*. PhD thesis, 1992.

[3] Rafik Henia, Arne Hamann, Marek Jersak, Razvan Racu, Kai Richter, and Rolf Ernst. System level performance analysis - the symta/s approach. 2005.

[4] Marek Jersak. *Compositional Performance Analysis for Complex Embedded Applications*. PhD thesis, Technical University of Braunschweig, 2004.

[5] Jukka Mki-turja and Mikael Nolin. Tighter response-times for tasks with offsets. 2004.

[6] M. Rudorfer O. Scheickl. Automotive real time development using a timing-augmented autosar specification. *BMW, Munich*.

[7] Jun Sun, Riccardo Bettati, and Jane W. s. Liu. An end-to-end approach to schedule tasks with shared resources in multiprocessor systems. 1994.

[8] Ken Tindell and John Clark. Holistic schedulability analysis for distributed hard real-time systems. *Microprocess. Microprogram.*, 40(2-3):117–134, 1994.