Journal of Systems Architecture 80 (2017) 104–113

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

End-to-end timing analysis of cause-effect chains in automotive embedded systems

Matthias Becker[*,a], Dakshina Dasari[b], Saad Mubeen[a,c], Moris Behnam[a], Thomas Nolte[a]

[a] MRTC / Mälardalen University, Västerås Sweden
[b] Robert Bosch GmbH, Renningen, Germany
[c] Arcticus Systems AB, Järfälla, Sweden

A B S T R A C T

Automotive embedded systems are subjected to stringent timing requirements that need to be verified. One of the most complex timing requirement in these systems is the data age constraint. This constraint is specified on cause-effect chains and restricts the maximum time for the propagation of data through the chain. Tasks in a cause-effect chain can have different activation patterns and different periods, that introduce over- and under-sampling effects, which additionally aggravate the end-to-end timing analysis of the chain. Furthermore, the level of timing information available at various development stages (from modeling of the software architecture to the software implementation) varies a lot, the complete timing information is available only at the implementation stage. This uncertainty and limited timing information can restrict the end-to-end timing analysis of these chains. In this paper, we present methods to compute end-to-end delays based on different levels of system information. The characteristics of different communication semantics are further taken into account, thereby enabling timing analysis throughout the development process of such heterogeneous software systems. The presented methods are evaluated with extensive experiments. As a proof of concept, an industrial case study demonstrates the applicability of the proposed methods following a state-of-the-practice development process.

1. Introduction

Automotive systems are getting complex with respect to traditional components like the Engine Management System (EMS) as well as modern features like assisted driving. While the increase in the EMS complexity is attributed to newer hybrid engines and stricter emission norms, assisted driving requires the perfect convergence of various technologies to provide safe, efficient and accurate guidance. This has led to software intensive cars containing several million lines of code, spread over up to hundred Electronic Control Units (ECU) [1].

Given this complexity, over the last decades, standards like AUTOSAR [2] have been proposed in order to conceive a common platform for the development of automotive software. These standards allow software components provided by different suppliers to be integrated on the same ECU, since they provide for a hardware agnostic software development. Such robust interfaces enable designers to design software in the early stages without knowledge of the concrete hardware platform on which it will be eventually executed. Thus, during the development it is often not known which other applications share the same execution platform.

Most of these automotive applications typically have strict real-time requirements – it is not only important that a computation result is correct, but also that the result is presented at the correct time. In addition to individual timing requirements on the response times of the tasks (i.e. the deadline of a task corresponds to the task's response time), these applications often have timing requirements on the end-to-end functionality of the task chains, so-called *end-to-end timing requirements*. Note that the task chains are commonly found in single-node as well as distributed real-time systems. On the one hand, the end-to-end timing requirement can be specified as an end-to-end deadline on a task chain, which corresponds to the end-to-end response time of the task chain. The end-to-end response time of a task chain is equal to the response time of the last task in the chain. The end-to-end deadline is considered satisfied as long as the response time of the last task in the chain is less than or equal to the specified deadline. It does not matter if the data from the input of the chain is transferred to the output of the chain within the specified end-to-end deadline or not. On the other hand, the end-to-end timing requirement on a task chain can also be specified by means of various timing constraints such as the age constraint [3–7]. The age constraint is specified on the data propagation

* Corresponding author.
E-mail addresses: matthias.becker@mdh.se (M. Becker), dakshina.dasari@de.bosch.com (D. Dasari), saad.mubeen@mdh.se (S. Mubeen), moris.behnam@mdh.se (M. Behnam), thomas.nolte@mdh.se (T. Nolte).

http://dx.doi.org/10.1016/j.sysarc.2017.09.004
Received 10 February 2017; Received in revised form 14 June 2017; Accepted 20 September 2017
Available online 27 September 2017

through a chain of semantically related tasks, where it specifies the maximum time from reading an input value by the first task of the chain, until a corresponding output value is last produced at the end of the chain. It is of utmost importance that the data from the input of the chain is transferred to the output of the chain within the specified constraint. It is of utmost importance that the data from the input of the chain is only affecting the output of the chain within the interval specified by the constraint.

Many design decisions, such as task activations and communication semantics, have direct influence on the data age. Thus, bounding the data age of a chain early during the design process can potentially avoid costly software redesigns at later development stages. The analysis gets complex as a chain may consist of tasks with different periods leading to over- and under-sampling situations. Such systems are called multi-rate systems. Most of the available analysis methods for such systems focuses on the implementation level where detailed scheduling information is available [8] and thus they are not applicable during early phases. In [9], a generic framework to calculate the data age of a cause-effect chain is presented, which targets single processor systems and is agnostic of the scheduling algorithm used.

Contributions: In this paper, we extend our earlier work on analyzing end-to-end delays among multi-rate effect chains [9] to utilize the information available at different levels of timing information. Specifically, we highlight the generic nature of the framework by showing how varied levels of information can be used to compute the maximum data age of systems with following characteristics:

1. Different activation patterns between tasks of one cause-effect chain, where individual tasks can be event triggered or periodically triggered.
2. Knowledge of the communication semantic: We extend the analysis to incorporate the explicit communication semantics as well as the Logical Execution Time (LET). Both being communication paradigms commonly used in the automotive domain.
3. Knowledge of offsets: The analysis for systems without knowledge of the schedule is extended to allow for task release-offset specifications.
4. Knowledge of the scheduling algorithm (like Fixed Priority Scheduling (FPS)): Most ECUs utilize operating systems which schedule tasks based on FPS. This allows to utilize existing analysis for such systems to determine worst-case response times of the individual tasks. It is then shown how the concepts of the analysis can be adapted to account for this information.
5. Knowledge of the exact schedule: Similar to most of the existing end-to-end delay analyses, we show how the exact schedule can be used to determine the exact delays with low computational overheads.

Finally, we compare these different scenarios with extensive evaluations, considering i) the tightness of the computed bounds and ii) the computation time for the analysis. We show that increased system information during the design process can thus be used to obtain tighter end-to-end latencies. An industrial case study is performed that demonstrates the applicability of the proposed methods in a state-of-the-practice tool-chain.

2. Related work

The Timing Augmented Description Language (TADL2) [4] was conceived out of the need for a timing model in the automotive domain, and is the basis for the timing constraints that can be defined in AUTOSAR [3] and EAST-ADL [5]. The end-to-end timing constraints found in these automotive multi-rate systems were first discussed in [10]. Here, the authors describe the different design phases and link them to EAST-ADL [5] and AUTOSAR [2]. An increased level of system knowledge during the consecutive design phases is outlined. Note that these types of end-to-end delays focus on data propagation between independently triggered tasks, in contrast to the end-to-end response time considering the first response at the end of a chain of tasks, where tasks may trigger each other [11].

A method to compute the different end-to-end delays of multi-rate cause-effect chains is presented in [8]. In addition, the authors relate the reaction delay to "*button to reaction*" functionality and the maximum data age delay to "*control*" functionality. In this work the focus lies on the maximum data age and hence on control applications.

A model-checking based technique to compute the end-to-end latencies in automotive systems is proposed in [12]. The authors generate a formal model based on the system description which is then analyzed.

The end-to-end timing analysis in an industrial tool suite is discussed in [6]. Two different activation methods are discussed; trigger chains, where a predecessor task triggers the release of a successor task, and data chains, where tasks are individually triggered and hence over- and under-sampling may occur. These activation patterns are supported by several modeling technologies including the AUTOSAR standard [2]. The trigger activation pattern is also supported by middleware approaches, e.g., a distributable thread in Real-Time CORBA [13] closely resembles the trigger chain. In this work we consider the chains that can have any activation pattern.

End-to-end delays in heterogeneous multiprocessor systems are analyzed in [14]. Ashjaei et al. [15] propose a model for end-to-end resource reservations in distributed embedded systems, and also present the analysis, based on [8], for end-to-end delays under their model.

Additionally, several industrial tools implement the end-to-end delay analysis for multi-rate effect chains [16–19]. However all of the discussed works require system information which is only available at the implementation level. In [9], a scheduling agnostic end-to-end delay analysis for data age is described, where only information about the tasks of the cause-effect chains is required. Additionally, this work shows how to add job-level dependencies to a task set, such that the data propagation between tasks in an effect-chain is restricted in a way that end-to-end delay constraints are met irrespective of the scheduling decisions.

The principle behind the job-level dependencies can be related to the *rate transition operation* of PRELUDE [20], which is a synchronous language for multi-rate real-time systems, based on the principles of LUSTRE [21]. LUSTRE is addressed in several works, [22] addresses such systems under fixed-priority scheduling [23], and [24] addresses systems under online priority-based scheduling. While these works address single-core systems, many-core target platforms are considered in [25,26].

The LET communication paradigm was introduced with the time triggered programming language Giotto [27,28]. Decoupling the execution and communication provides benefits for various areas of embedded systems where predictability and dependability are of most importance [29], such as the automotive industry [30].

In this work, we extend the results presented in [9], and show that by augmenting information available during the different design phases, we can analyze the maximum data age with decreasing degree of pessimism. It is further shown how the existing framework can be used for different trigger schemes and communication paradigms. Hence, the input for the framework is provided in such a way that the underlying mechanisms and related works can be leveraged, while the set of systems that can be represented is significantly enlarged.

2.1. Relation to authors' previous work

This work builds up on the timing analysis for the data age which was presented in [9], where early timing analysis is used to synthesize a partial ordering on the task's jobs to satisfy the corresponding data age constraints. This then significantly eases the synthesis process.

In [31] we extend this analysis by observing that adjustments of the read- and data-intervals can reflect the different levels of system