Extending EAST-ADL for Modeling and Analysis of System's Resource-Usage

Raluca Marinescu and Eduard Paul Enoiu
Mälardalen Real-Time Research Centre (MRTC)
Mälardalen University
Västerås, Sweden
Email: {raluca.marinescu, eduard.paul.enoiu}@mdh.se

***Abstract*—EAST-ADL is an architectural description language dedicated to automotive embedded systems design, with focus on structural and functional modeling. The current architectural notations lack support for modeling and analysis of resource-usage, and therefore it is not possible to reason about resource requirements. In this paper, we describe our work towards filling the gap between EAST-ADL language and formal modeling and analysis of system's resource usage, by extending the EAST-ADL language with embedded resources, such as storage, energy, communication and computation. To formalize this approach and provide a basis for rigorous analysis, we show how to analyze EAST-ADL models using the framework of priced timed automata and weighted CTL. We report our experiences from applying this approach for integrating resource-wise analysis into EAST-ADL.**

***Keywords*-EAST-ADL; priced timed automata; resource; analysis;**

I. Introduction

The complex architecture of automotive Embedded Systems (ES) and limited software and hardware resources make the resource-wise early-analysis and verification of system development, at multiple abstraction levels, very desirable [6]. EAST-ADL [7] is an architecture description language for modeling and development of automotive ES, covering engineering requirements, vehicle functions, software and hardware functionality, timing constraints, and other related information. To ensure high reliability for such applications, dependable components have to be analyzed to meet the requirements, subject to some resource constraints. Due to the absence of resource modeling notations in EAST-ADL, allocations for components can not be analyzed and refined at earlier phases of design.

The issue of solving such problems may be resolved by including resource-dependent information in EAST-ADL language, and adjust resource allocation to system changes, whenever there are constraints that have to be met. We are interested to assess how any design decision in EAST-ADL affects the system's overall resource consumption. This includes finding suitable analysis goals and appropriate trade-offs between different system configurations. The systematic analysis of EAST-ADL models enhanced with support for describing resource consumptions should provide the system designer with insights on the system′s resource-wise behavior and the feasibility of the design, with regard to resource analysis, especially if the provided resource budget is known.

In this paper, we propose a modeling extension to the current EAST-ADL language with associated abstract resource information. Next, we integrate this extension with a formal model, which supports resource analysis techniques for performing quantitative consumption analysis. Also, we show how analysis goals can be formalized and reasoned about.

Our approach combines EAST-ADL with an abstract behavioral model that supports resource-wise analysis. The main contributions of this paper are:

- an EAST-ADL language extension towards abstract resource information representation,
- an analysis framework for predicting resource usage and optimizing resource utilization, intended to narrow the gap between EAST-ADL and formal models, e.g., Priced Timed Automata (PTA), which is the formal framework for analysis.

The paper is organized as follows. Section II briefly overviews EAST-ADL and PTA framework. Section III introduces our approach towards modeling of resources extension, and the modeling approach for functional specification in EAST-ADL. Section IV describes our analysis method. Next, we apply our approach on the Brake-By-Wire case study in Section V. In Section VI we compare to related work, before concluding the paper in Section VII.

II. Preliminaries

A. EAST-ADL

EAST-ADL is an architecture description language refined and aligned with AUTOSAR automotive standard [14]. It is intended to support the development of automotive ES, by capturing modeling related engineering information.

The language defines five abstraction levels. The levels reflect different views and details of the architecture, from a higher level of abstraction down to components in hardware and software. Vehicle features are modeled at the `Vehicle Level`, the highest level of abstraction. These features are then refined at the `Analysis Level`, where a complete representation of the abstract functional definition of features in system context is modeled. The details for

978-0-7695-4758-9/12 $26.00 © 2012 IEEE
DOI 10.1109/COMPSACW.2012.99

functional definition of software, including elementary decomposition, are introduced at the Design Level. The Implementation Level describes reusable code and AUTOSAR compliant software and system configuration for hardware deployment [7].

B. Priced Timed Automata

The PTA framework [5] is an extension of timed automata [2] with prices on locations and transitions, in which the accumulation of prices is represented by continuous variables.

Let X be a finite set of clocks and $B(X)$ the set of guards, which are finite conjunctions of atomic guards of the form $x \bowtie n$, where $x \in X$, $n \in \mathbb{N}$, and $\bowtie \in \{<, \leq, =, \geq, >\}$.

A (Linearly) PTA over clocks X and actions Act is a tuple (L, l_0, E, I, P) where L is a finite set of locations, l_0 is the initial location, $E \subseteq L \times B(X) \times Act \times P(X) \times L$ is the set of edges, $I : L \rightarrow B(X)$ assigns invariants to locations, and $P : (L \cup E) \rightarrow \mathbb{N}$ assigns costs to both locations and edges. In the case of $(l, g, a, r, l') \in E$, we write $l \xrightarrow{g,a,r} l'$.

The semantics of PTA is defined as a transition system over states (l, u), with the initial state (l_0, u_0), where u_0 assigns all clocks in X to zero. There are two kinds of transitions:

(i) *delay transitions*: $(l, u) \xrightarrow{d,p} (l, u \oplus d)$, where $u \oplus d$ is the result obtained by incrementing all clocks of the automata with the delay amount d, and $p = P(l) * d$ is the cost of performing the delay, and

(ii) *discrete transitions*: $(l, u) \xrightarrow{d,p} (l', u')$, corresponding to taking an edge $l \xrightarrow{g,a,r} l'$ for which the guard g is satisfied by u. The clock valuation u' of the target state is obtained by modifying u according to updated r. The cost $p = P((l, g, a, r, l'))$ is the priced associated with the edge.

A trace σ of a PTA is a sequence of delays, actions, and transitions:
$\sigma = (l_0, u_0) \xrightarrow{a_1,p_1} (l_1, u_1) \xrightarrow{a_2,p_2} ... \xrightarrow{a_n,p_n} (l_n, u_n)$, where the cost of performing σ is $\Sigma_{i=1}^{n} p_i$.

Properties of PTA are expressed as logical formulae in the Weighted Computational Tree Logic (WCTL) [10]. The syntax is given by the following grammar:

$WCTL \ni \phi ::= true \mid a \mid \neg\phi \mid \phi \vee \phi \mid E\phi U_{P \bowtie c}\phi \mid A\phi U_{P \bowtie c}\phi$,
where a is an atomic proposition, P is a cost function, c ranges over $\mathbb{N}$ and $\bowtie \in \{<, \leq, =, \geq, >\}$, A and E are the universal and existential quantifiers, $U_{P \bowtie c}$ is the "until" temporal modality, and the temporal operators <> and [] are derived in the usual way.

A network of PTA $A_1 \parallel ... \parallel A_n$ can be expressed as a composition of n PTA over X and Act, synchronizing on actions (i.e., $b!$ is complementary with $b?$) and using shared variables that can be used in guards and transitions.

III. EXTENDING EAST-ADL WITH RESOURCE MODELING

Automotive ES are constrained by their limited software and hardware resources. It is crucial that architectural languages facilitate formal modeling and analysis of resource-usage. EAST-ADL is not formalizing a set of allocation-specific information, and specification of resource properties attached to allocations is missing. Due to such arguments it is desirable to augment EAST-ADL language with resource requirements. In our context, resource-aware architecture analysis relates to specific analysis goals used to predict resource usage and optimize resource utilization.

This paper discusses the abstract EAST-ADL functional models of a system. Specifically, on the Design Level, the system is described by a Functional Design Architecture (FDA), which is composed of a number of interconnected Function Prototypes (F_P), where each prototype is an instantiation of Function Type (F_T). In EAST-ADL functional modeling, systems in FDA are built from interconnected function blocks with well-defined interfaces consisting of a set of input and output function ports, respectively. For the presented work, the user specifications are clearly described from the structural, behavioral, and timing EAST-ADL packages [7].

We propose a notation extension that enables modeling and analysis of a set of resources on top of which an EAST-ADL application may be allocated, taking into account EAST-ADL functional specifications. In order to annotate and reason about resource usage of EAST-ADL models, we need a semantic extension of the model and its behavior. At the structural level we need to consider resources as part of our extension of the EAST-ADL language in order to obtain resource awareness at architectural level. At the behavioral level we can use PTA as the framework needed for the formal analysis of the corresponding models and the resource consumption.

A. Resource Notation Extension

A basic F_P corresponds to a basic intermediate functional block with a model that can capture the behavior of the associated F_T annotated with resource consumption information. The internal computation of an F_P starts with reading all input flow ports. This internal input data is used together with other functional information during the F_P execution, before writing the variables to the output flow ports.

In order to represent in the same FDA diagram, the architecture with respect to timing, behavior, and resources of the required functional blocks, we have conceptually extended the EAST-ADL notation with resource constraints. We consider FDA to be mapped onto the virtual architecture as an abstraction of the targeted platform. For the elements of the virtual architecture, resources like memory, CPU, and bandwidth, are available.

Consider an EAST-ADL model composed of a number of functional blocks that need to be represented as elements of the selected architectural structure specification, and extended with information from a package named Resource Function. This extension is intended to be used, as it is, wherever a resource is necessary to be coupled to an F_P, and is basically a specification, as depicted in Fig. 1. The resource modeling can contribute to FDA analysis, as this level of abstraction is dealing with functional modeling. The Constraint is a container of Resource Function. It enables to regroup the resources assigned to functional blocks in a particular context, on which functional modeling can be applied. The collection of functional resources can be extended across the EAST-ADL abstractions levels on functional modeling. We assume that the corresponding functional model is consuming the prototype of resources. Such consumptions can be expressed as the rate at which the functional block consumes the resources in time. Our goal is to analyze various scenarios of the system's resource usage, and calculate, e.g., the minimum or maximum amount of resources for correct behavior. In order to be able to analyze this extension of the EAST-ADL model, we need a semantic implementation and translation to a formal representation.

We define a minimal structural model implementation, an intermediate model, from which we can derive the constructs of the EAST-ADL language, and derive the translation to the PTA formalism. One step towards support for formal analysis is the integration of this intermediate model, by which functional behavior, resource consumption and timing can be addressed in a single modeling formalism. Using these relations, we can describe all constructs in our assumed EAST-ADL model on FDA. A simple one-to-one mapping rule between structural entities is not sufficient though. Several parameters need to be handled in the translation process.

IV. Analyzing resource-aware EAST-ADL Models

We consider the cost of using the resources in a component-based feasibility, optimal, and worst case resource consumption analysis. The cost criteria may involve memory, bandwidth, and CPU usage. We use PTA as the formalism to model the EAST-ADL functional blocks in FDA with respect to a given resource cost criteria. Also, the notion of time is essential to our model (i.e. the longer the system executes, the more resources it consumes). This makes PTA the model of choice to reason about resource usage at analysis and design level in EAST-ADL. If we consider the resource consumption annotations as cost variables, and we employ the PTA framework as the underlying formal model representation, then the translation of a model that complies with the EAST-ADL language extended with resource annotations implies that:

- for each non-elementary F_T, we define a network of PTA,
- for each F_C and flow ports for passing control, we define synchronization channels in PTA,
- for each elementary F_T, we create a PTA model.

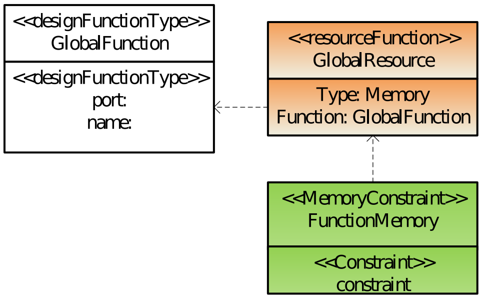

Figure 1. Resource notation extension for a function type

The obtained PTA represents the execution behavior with functional, timing, and resource information. The internal application behavior corresponds to the actual behavior mentioned in the model implementation, with available timing and content of functional blocks. The formal translation of the EAST-ADL language annotated with resources into a network of PTA and the associated tool are subject to future work.

A. Feasibility analysis

Different types of analysis can be used for resource consumption at architectural level. FDA *feasibility analysis* can be used to verify if the available resource provided in the implementation level are matched by the resources consumed during the actual execution.

We consider the linearly-priced resource consumption adequate for modeling and analysis at the architectural level, before the AUTOSAR implementation. The obtained behavioral PTA model of the system corresponds to a TA whose locations and edges correspond to resources being consumed. We consider that the price-per-time-unit of every resource is constant, and the price function is linear. This resource modeling approximation suffices for our purposes, since we are interested in high-level analysis of early EAST-ADL design models. If we desire to make more accurate predictions, it is always possible to use benchmarks to operate on the target platform and to use the measured values.

If we consider the individual resource consumption on the PTA model of the system, one can use feasibility analysis to verify if a certain resource consumption on all possible behaviors stays within the available resource amounts provided by the platform. If we consider the PTA model to be the semantic translation of our intermediate model and *cost* is the accumulated resource consumption, the verification goals can be formalized as WCTL properties used for checking our model:

$$A <> \ _{cost \leq max}\ p \tag{1}$$

$$E <> \ _{cost \leq max}\ p \tag{2}$$

The above verification goals are specified as liveness proper-

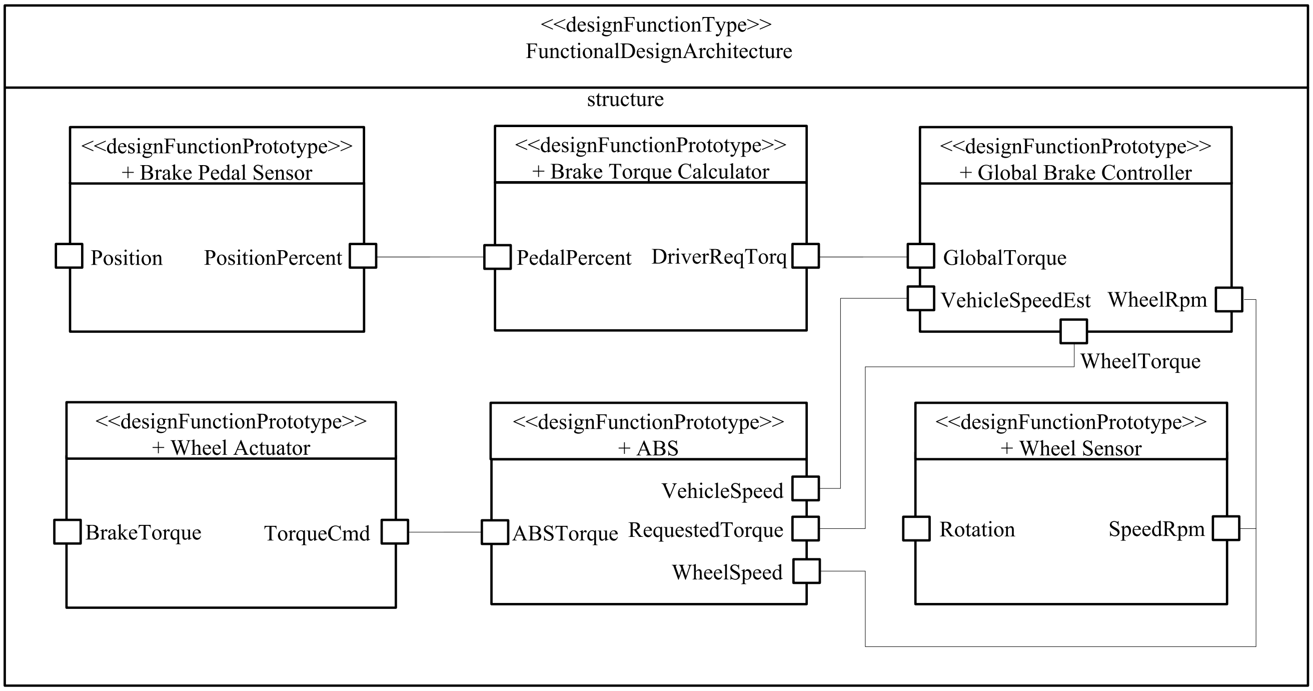

Figure 2. Brake-by-Wire functional model in EAST-ADL

ties (1) and as reachability properties (2) [15]. Property (1) requires that for all execution paths, the p location in the PTA model is eventually reached with less or equal max cost. Property (2) states that there is a path in which the p location in the PTA model is reached within a maximum resource cost max.

B. Optimal resource consumption

One can use optimal and worst-case resource consumption analysis on the PTA model, which computes the cost of the trace that will eventually reach a certain behavior. This problem reduces to minimizing/maximizing the resource cost function such that a property is satisfied. Optimization of resources in our case is actually the optimal reachability problem proposed by Larsen and Rasmussen [10] (minimum/maximum cost for reaching a given behavior or timing goal).

The tool used for verifying optimal resource consumption properties is UPPAAL CORA [1], where one could check the ($E <> \ p$) property, while the tool calculates the optimal resource cost to reach the p location of the corresponding PTA. The current version of UPPAAL CORA tool supports only minimal reachability analysis (for optimal resource cost), but theoretically, feasibility checks as maximal reachability problems are also proven decidable [9]. Alternatively, this information can be translated back into the EAST-ADL resource extension, and later used for further resource analysis by checking whether the resource demands of some functional blocks are smaller than the available ones.

V. EXAMPLE: BRAKE-BY-WIRE

We have checked the applicability of our approach on a `Brake-By-Wire` (BBW) system modeled in EAST-ADL. The case study is based on the use case provided by Volvo Technology within the MBAT project [2].

Fig. 2 depicts a simplified EAST-ADL diagram, which presents the BBW system at the design level in the `Functional Design Architecture`. The functionality of the system can be divided between sensors and actuators F_Ps, and computations F_Ps. The `Brake Pedal Sensor` (BPS) transforms the raw data (the voltage that is related to the pedal angle) into a percentage of a maximum value of the brake force. In case of the `Wheel Sensor` (WS), the raw data corresponds to the actual wheel speed. The `Wheel Actuator` (WA) is modeled as a F_P, which is computing the brake force. The computation is divided between two F_Ps: the `Brake Torque Calculator` (BTC) computes the requested brake force, which is used as input in the `Global Brake Controller` (GBC) that calculates a basic brake force on each of the wheels. The ABS controls the wheel braking in order to prevent locking the wheel, based on the slip rate value.

[1] For more details on UPPAAL CORA we refer the reader to http://people.cs.aau.dk/ adavid/cora/

[2] For more details on ARTEMIS MBAT consortium visit the website http://www.mbat-artemis.eu/

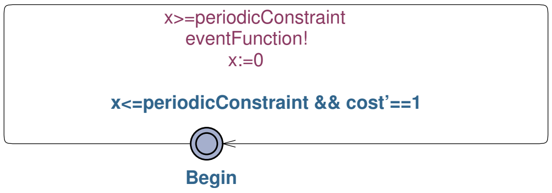

Figure 3. The PTA model of the periodic constraint for the ABS function prototype

In the BBW model, we assume the use of one available resource: memory or processor. We assume that the processor resource uses one tick per instruction, and the memory is static, the memory being allocated when a F_P is triggered, and released as soon as the F_P is exited.

A. A PTA model of BBW

We have analyzed the BBW system, as a network of PTA models, in UPPAAL CORA. The PTA models of the event Function, and the ABS Function Prototype are shown in Fig. 3 and 4. For analysis purposes and simplicity, we are showing here only two PTA from the BBW system shown in Fig. 2. The eventFunction synchronizes with the ABS model every Periodic Constraint through the channel eventFunction. The ABS PTA has eight locations including: Start, calculateSlipRate, TorqueCmd, BrakeTorque, and ABS. The synchronization with Global Brake Controller is modeled using channels wheelSpeed and vehicleSpeed. The ABS controls the wheel braking in order to prevent locking the wheel, based on the slipRate variable. The TorqueCmd location is reached if slipRate is greater than 20, implying that the brake actuator is released and no brake is applied, otherwise the requested brake torque is used. Consequently, the ABS enters location BrakeTorque, and jumps back to location Start.

B. Resource Analysis of the BBW model

In the PTA model analysis, we have a constant cost of resources. The cost of resource usage is influenced by the individual behavior of the PTA, as the utilized resource on each edge. For the current work in UPPAAL CORA, the PTA models can handle only monotonically increasing cost functions. In our BBW example, we consider a cost function within a single cost variable, with possible weights, representing the relative importance of each resource.

As described previously [8], EAST-ADL models enriched with TA behavior can be verified for safety or properties like the following:

$$A[\,](ABS.slipRate > 20\ imply\ (ABS.brake == 0))$$

This property checks one of the system's functional requirements, which is related to the slip rate. It verifies the following functionality: in case the slip rate variable exceeds 20, the brake actuator is released and no brake is applied.

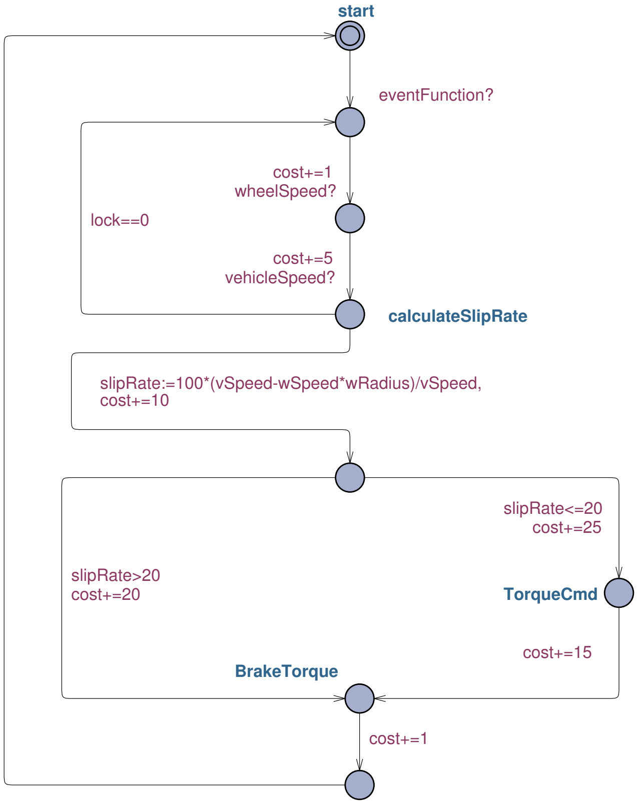

Figure 4. The PTA model of the ABS function prototype

For the presented work, we are interested to analyze the optimal cost reachability problem for computing the minimum cost on a corresponding trace produces by UPPAAL CORA. For example, one can analyze the lowest resource usage on specific sequence of braking scenarios, that results in a minimization of the cost function. This information can be used in the design and analysis of the BBW EAST-ADL system for finding interesting execution behaviors with respect to different resource usage scenarios. As an illustration of this technique, we check for an optimal resource trace satisfying the property:

$$E <> ABS.BrakeTorque,$$

that is, a trace in which the torque command is eventually sent to the wheel, based on the calculated slip value. Consequently, UPPAAL CORA has generated the execution in which the brake actuator is released, no brake is applied, with the cheapest trace with respect to total resource usage.

VI. Related Work

Early analysis requires work on more abstract descriptions of the expected resource usage. Such abstract descriptions have to state more than how many resources are needed, but they should include information about the time needed for the resources to be available. Some work has been done and proposed to tackle the analysis of embedded resources [3] using UML notations to complement architecture description languages.

MARTE profile [1] is one of the major efforts on modeling of real-time embedded systems and their non-functional properties. MARTE enables analysis with regard to resource allocation to enable early scheduling analysis and provides different packages which serve to represent the type of resource requirements [11]. This packages are distinguishing two kinds of processing resources (e.g. execution nodes) and communication resources (networks and buses). MARTE supports the separation of concern at different abstraction levels [13]. Despite the fact that currently MARTE consists of a sub-profile for performance, it does not offer formal analysis with regard to an abstract resource, as our approach does. Also, the alignment of MARTE and EAST-ADL is still an ongoing process that could be enriched with our approach in order to deal with different resource analysis problems.

Last, but not least, research has been devoted to provide the generic formal foundations [16] to establish a process of modeling and analysis of resources in component-based systems from low-level code resource estimates [12] to higher level UML and formal approaches [4]. Nevertheless there is a a gap between EAST-ADL and early resource usage analysis. In such a case we focus on using abstract descriptions of expected resource usage at architectural level in EAST-ADL, and provide the designer with analysis means for optimizing the overall resource usage of a system, with respect to cost of the available resources.

VII. Conclusion

In this paper, we have provided means for the systematic modeling and analysis of EAST-ADL models with support for resource annotations. We describe an EAST-ADL extension for resources that can be allocated, consumed and released. Obvious type of resources that can be modeled include CPU, memory, and bandwidth. In order to describe usage of resources in an EAST-ADL model, we annotate resources on the behavioral description of the functional blocks. We have identified two important goals: feasibility analysis and optimal resource analysis. To support analysis, we have shown by a BBW example how EAST-ADL models annotated with resources can be analyzed using priced-timed automata framework. The studied example is a Brake-by-Wire system that consumes resources. The system is architecturally modeled in EAST-ADL, and uses PTA to model function, timing, and resource usage of the included functional blocks. To analyze the optimal resource usage, we model the resources as costs, and use a network of priced timed automata for performing analysis in the UPPAAL CORA tool.

As future work, we plan to derive PTA models from UML state diagrams used natively in EAST-ADL, that can be augmented with resource performance information.

Acknowledgment

The research leading to these results has received funding from the ARTEMIS Joint Undertaking under grant agreement number 269335 and from VINNOVA, the Swedish Governmental Agency for Innovation Systems.

References

[1] Huscar Espinoza A and Hubert Dubois A. A general structure for the analysis framework of the uml marte profile. 2008.

[2] Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183 – 235, 1994.

[3] Hany H. Ammar, Vittorio Cortellessa, and Alaa Ibrahim. Modeling resources in a uml-based simulative environment. 2001.

[4] Steffen Becker, Heiko Koziolek, and Ralf Reussnerl. The palladio component model for model-driven performance prediction. *Journal of Systems and Software*, 2009. Software Performance - Modeling and Analysis.

[5] Gerd Behrmann, Ansgar Fehnker, Thomas Hune, Kim Larsen, Paul Pettersson, Judi Romijn, and Frits Vaandrager. Minimum-cost reachability for priced time automata. In *Hybrid Systems: Computation and Control*, Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2001.

[6] Manfred Broy. Challenges in automotive software engineering. In *Proceedings of the 28th international conference on Software engineering*, pages 33–42, 2006.

[7] The ATESST2 Consortium. East-adl profile specification. www.atesst.org, 2010.

[8] Eduard Paul Enoiu, Raluca Marinescu, Cristina Seceleanu, and Paul Pettersson. Towards the analysis and verification of east-adl models using uppaal port. Technical Report, Mälardalen University, March 2012.

[9] K.G. Larsen and J.I. Rasmussen. Optimal reachability for multi-priced timed automata. *Theoretical Computer Science*, 390(2-3):197–213, 2008.

[10] Kim Larsen and Jacob Rasmussen. Optimal conditional reachability for multi-priced timed automata. In *Foundations of Software Science and Computational Structures*, Lecture Notes in Computer Science. 2005.

[11] F. Mallet, M.-A. Peraldi-Frati, and C. Andre. Marte ccsl to execute east-adl timing requirements. In *IEEE International Symposium on Real-Time Distributed Computing 2009*, march 2009.

[12] Johan Muskens and Michel Chaudron. Prediction of run-time resource consumption in multi-task component-based software systems. In *Component-Based Software Engineering*, Lecture Notes in Computer Science. 2004.

[13] Dorin Petriu and Murray Woodside. An intermediate metamodel with scenarios and resources for generating performance models from uml designs. *Software and Systems Modeling*, 2007.

[14] Tahir Naseer Qureshi, DeJiu Chen, Henrik Lönn, and Martin Törngren. From east-adl to autosar software architecture: a mapping scheme. In *Proceedings of the 5th European conference on Software Architecture*. Springer-Verlag, 2011.

[15] Cristina Seceleanu, Aneta Vulgarakis, and Paul Pettersson. Remes: A resource model for embedded systems. In *In Proc. of the 14th IEEE International Conference on Engineering of Complex Computer Systems*. IEEE CS, June 2009.

[16] Aneta Vulgarakis and Cristina Seceleanu. Embedded systems resources: Views on modeling and analysis. In *1st IEEE International Workshop On Component-Based Design Of Resource-Constrained Systems*, July 2008.