2012 IEEE 17th International Conference on Engineering of Complex Computer Systems

A timing model for specifying multi clock automotive systems

The Timing Augmented Description Language V2

M.-A. Peraldi-Frati, A. Goknil, J. DeAntoni
AOSTE Project
UNS-I3S-INRIA
Sophia Antipolis, France
map@unice.fr, {arda.goknil julien.deantoni}@inria.fr

J. Nordlander
Chalmers University of Technology
412 96 Göteborg
Sweden
johan.nordlander@ltu.se

Abstract— Precise timing constraint modeling and analysis is a key point for the correct development of automotive electronics. EAST-ADL and AUTOSAR has been adopted as standards in automotive industry. These standards have recently adopted TADL (Time Augmented Description Language), a timing model for expressing timing constraints. Its current use highlighted different issues, mainly concerning the integration of parameterized multi rate and multi-clock systems. This paper presents new extensions, aligned on AUTOSAR and EAST-ADL, to solve these issues: a support for symbolic timing expression including multi time base description and complex timing constraints. These extensions are applicable at different abstraction levels during design and enable precise modeling of the multi clock characteristics of distributed systems together with parameterized timing expressions. This work has been conducted in the ITEA TIMMO-2-USE project.

Keyword: Timing Requirements, Multi-clock systems modeling, Model-driven development, Timing analysis, EAST-ADL2, AUTOSAR

I. INTRODUCTION

Complexity and number of embedded software functionalities in automotive systems are drastically increasing over the years. In a vehicle, these functions are generally allocated into multiple Electronic Control Units (ECU) interconnected with CAN [1] or Flexray [2]. Some of these functions such as adaptive cruise control (ACC), Anti Blocking System (ABS) or Ignition Control System (ICS) are time critical and must be safe, *i.e.* their timing constraints must be fulfilled and the overall behavior of the system have to be proven to be correct.

For the last 10 years, the automotive industry has fully invested in the AUTOSAR standard [3] and EAST-ADL [4] as supports for the development of such embedded systems. These standards provide supports for a multi level development process based on different abstraction levels allowing a separation of concerns between hardware and software parts. Additionally, it helps abstracting the hardware by making intensive use of common interfaces for buses, operating systems and reusable components.

Both the new releases of AUTOSAR (V4) [5] and EAST-ADL (V2) [6] have adopted the timing model proposed in the Timing Architecture Description Language (TADL) [7]. TADL allows the specification of timing requirements on top of AUTOSAR and EAST_ADL models by defining classical timing characteristics of systems (duration, period, synchronization, etc.). However, TADL still lacks expressivity and is not able to capture important timing aspects such as:

- The integration of complex concepts of distributed systems such as multi rate and multi clock systems (a car software being distributed on different ECUs).
- The modeling of symbolic timing expressions, *i.e.* able to deal with bounded or unset parameters.
- The modeling of constraints that bridge the gap between hardware and software parts of a system.

Supporting these new features in EAST_ADL and AUTOSAR is a key point to enable the effective use of analysis tools all along the development process (i.e. from requirements to implementation). This is the main issue we want to cope with in the ITEA TIMMO-2-USE project in order to make automotive standards usable and effective. Consequently, we propose to extend TADL with an explicit notion of time base, and to represent the various temporal referentials used in an automotive design (clocks from different ECUs, motor position, etc). Of course the creation of relations between such referentials is also a part of the extensions. Additionally, all timing expressions are augmented with parameters, which can be free at the highest abstraction level and then progressively defined during the design refinement. As a result, a symbolic timing expression in TADL2 is possibly made of a suitable set of arithmetic operators mixing symbolic identifiers (not necessarily set variables) and referring to different time bases. One typical use of this feature is to capture unknown configuration parameters; another one is to relate constraints in different time-bases to each other. Inherent to this work is also the study of the allowable ranges for symbolic values that are dictated by a set of constraints. TADL2 proposes an

978-2-9541-8100-4/12 $26.00 © 2012 IEEE
DOI 10.1109/ICECCS.2012.5

alignment of these concepts on the current version of EAST-ADL2 and AUTOSAR.

The paper is organized as follows: section II presents a "Brake-By-Wire" system and its timing expressions to illustrate the needs in such domain. Section III gives our motivation for the new concepts in TADL2 w.r.t. the related existing approaches. Section IV is dedicated to the presentation of TADL2. We present the metamodel for symbolic and multi time base timing expressions with the associated concrete syntax. In Section V, we explain the TADL2 modeling environment. Section VI concludes the paper.

II. Example of a Brake-by-Wire Application

A distributed brake-by-wire application with anti-lock braking functionality illustrates our approach. The brake-by-wire application is one of the validator proposed by Volvo Technology in the TIMMO-2-USE project [8].

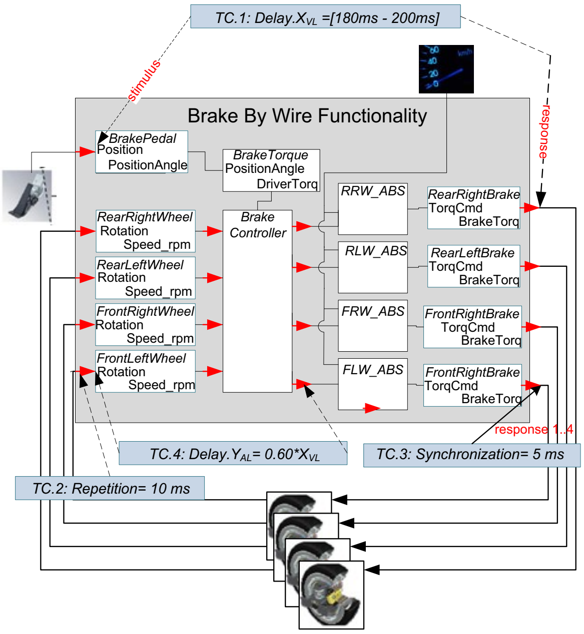

Figure 1. Brake-By-Wire functional view

The structural decomposition of the braking functionality is showed in Figure 1. The BBW is composed of two mains functions. First a brake controller reads wheel speed sensors and a brake pedal sensor. The brake controller computes the desired brake torque to be applied to the wheels. In addition to this basic brake controller functionality, a second function ABS (Anti blocking System) adapts the brake force on each wheel if the speed of one wheel is significantly smaller than the estimated vehicle speed. In this case, the brake force is reduced on that wheel until it regains speed that is comparable with the estimated vehicle speed. The ABS takes as inputs the sensors values on each wheel and the estimated vehicle speed.

Figure 1 gives examples of timing constraints (TC) applied to this functional description:

TC.1: A ***Delay*** *constraint X_{VL} is bounded with a minimum value of 180ms and a maximum value of 200ms. This delay is measured from brake pedal stimulus to brakes response. Here, activation of the brake pedal sensor is the stimulus and brake actuation is the response.*

TC.2: A Periodic acquisition of wheel sensors must be done with a ***Repetition*** *constraint of 10 ms.*

TC.3: The tolerated maximum ***Synchronization*** *constraint between first and last wheel brake actuation is 5 ms.*

TC.4: The ***Delay*** *constraint applied on sensor acquisitions and brake controller is a percentage of the initial time budget X_{VL}.*

In the design process based on EAST-ADL and AUTOSAR, the functional description is refined while passing different development levels. Figure 2 shows a complete view of BBW timing constraints that follows the functional decomposition through the levels.

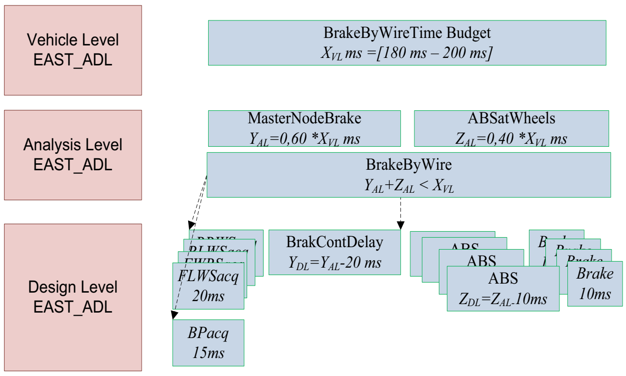

Figure 2. BrakeByWire Timing constraints

At the vehicle level, a timing constraint is assigned by the car maker to the different suppliers. The timing budget is X_{VL} ms at the vehicle level. An interval value of [180ms-200ms] is assigned to X_{VL}.

At the analysis level, this time budget is shared by sub-functions. Timing constraints are refined and associated with two main functions that cover sensors acquisition, brake controller, ABS and Brake Actuation components. This is called time segments and in this case time segment constraints are expressed as percentages of X_{VL}. We should be able to refer to a variable in the model and integrates this variable into a timing expression.

As X_{VL} is an interval, an additional timing constraint at the analysis level states that the sum of time segments should be less than X_{VL}.

A first analysis of these timing constraints induces requirements to be fulfilled by a time modeling language:

Req.1: Timing language should cover the expression of duration delay, periodic execution, and synchronization interval.

Req.2: Timing language should express timing constraints made of a suitable set of arithmetic operators mixing constant value and symbolic identifiers.

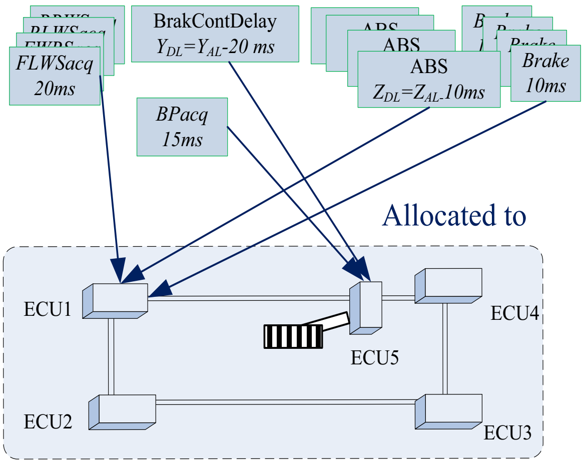

Figure 3. Hardware architecture and allocation of BBW functions

The hardware architecture and the allocation of BBW functions on this architecture are represented in Figure 3. The hardware platform consists of sensors/actuators and computing parts (five electronic control units connected by a communication bus). Each ECU has its own temporal reference (timebase), which is not necessarily (well) synchronized with the other one and the communication between them is still mainly asynchronous (despite the apparition of Time Triggered buses). Such potential drifts between timebases of computing hardware parts (ECU clocks) or latencies in communication parts (bus, memory access…) should be part of the design. In the BBW, associated timing constraints should state that:

TC.5: ECU5 has a drift of 0.02 millisecond for each second compared to the universal time.

TC.6: ECU5 timebase goes 2 times faster than timebases of ECU 1 to 4.

Timing constraints can also be linked to non computational hardware parts (i.e.: in an engine control system, ignition control shall be computed for each 720° measured on a crankshaft). In this case the timebase is the Crankshaft revolution.

Introducing timebases in timing constraint modeling is mandatory to cope with complex distributed systems such as multi rate and multi clock systems. The associated requirement for the language is the following:

Req.3: Timing constraints should refer to different timebases. Timebase should be of different types (i.e. measuring time, angle, distance or logical).

Time bases can be related to each other by either constant values or dynamic relations (example dependency between °CRK and engine round per minute speed). Expressing relationship between timebases is mandatory for building a global perception of time and ensuring a time safe cooperation over the platform. Req.4 expresses this need:

Req.4: Timebases should be related to each other by either constant values or dynamic relations which allows a global understanding of the timing behavior of a system.

III. MOTIVATION & RELATED WORK

Capturing behavioral and structural requirements at the early stage of a design becomes a mandatory requirement of modern process design flow for real-time embedded systems. Several projects such as ATTEST2 [3], AADL-OSATE [9], AUTOSAR [3] or even SysML [10] have produced significant results for the development of sophisticated safety critical systems by providing environments that allow the development process to start with high level modeling. These models focus on the system parts that are relevant for analysis while abstracting away the irrelevant details. In this context, extra-functional properties are more and more important at the model level. In the previous section we highlighted the need for a multi clock and parameterized temporal specification.

Many approaches use specific temporal annotations on a model to add the information required by scheduling analysis tools [11] [12] [13] [14]. These approaches provide temporal analysis on models but they are not abstract enough. They consider that all value are already fixed and known (i.e. they are suitable only to the very last steps of the development process) Therefore, they are not suitable for high level modeling and analysis of "distributed systems". These aspects have been highlighted in section II as mandatory in the automotive domain where several Electronic Control Units (ECU) communicate through buses.

The description of the temporal properties should allow the description of the possible loosely synchronizations to be amenable to correct analysis. In addition, automotive systems control physical parts such as an engine. Some temporal properties refer to these physical parts such as the position of the camshaft (see *Req.3* in section II).

The TIMMO-2-USE [8] project aims at overcoming these issues. The project proposes the Timing Augmented Description Language (TADL [15]) aligned on AUTOSAR to model temporal properties at the analysis and design phases of a development process. In TIMMO-2-USE project, a methodology to explore the timing features for a progressive and step by step system description has been proposed. Simulation tools [16] [17], connected to a TADL description, provide end-to-end delay scenario and deliver time stamps of tasks and functions calls. Figure 4 shows a partial view of the AUTOSAR metamodel. It provides the

TimingConstraint modeling element with lower/upper values expressed by the *TimeDuration* element where the value property is a constant. Additionally, it gives the possibility to select among different units and unit types. References to these units are selected in a hard coded enumeration (cseCodeType) that contains two implicit timebases from two physical dimensions: the chronometric time and the angular degree. It cannot be extended by the system/software architect without modifying the metamodel (i.e. the concepts of the tool itself). While these features greatly improved the modeling of automotive systems, many issues have still to be solved:

- AUTOSAR *TimingConstraint* is able to express totally defined requirements on a unique and implicit timebase. Timing constraints cannot be parameterized so that they can only be specified later in the development process. They are not amenable to cover complex arithmetic timing expressions as expected in *Req.2*.
- EAST-ADL and AUTOSAR have only two implicit timebases from two dimensions and cannot express any other timebases (distance, temperature, etc.). Consequently, they cover partially the needs expressed in *Req.3*.
- AUTOSAR allows modeling units of different nature (ms, s, °, etc.) by using the cseCode property. The relation between these units is a multiplication factor (cseCodeFactor) that should be expressed for each TimeDuration expression. The needs expressed in *Req.3* and *Req.4* for an explicit modeling of timebases and expressing the relations between them are not met by the current standard.

Alternatively to the previous approaches, academic researches proposed the use of *multiform* and *logical time* to deal with the previously exposed problems. Logical time has proved its benefits in several domains. It was first introduced by Lamport to represent the execution of distributed systems [18]. It has then been extended and used in distributed systems to check the communication and causality path correctness [19]. Logical time has also been intensively used in synchronous languages [20] [21] for its multiform nature. The multiform nature of logical time consists in the ability to use any repetitive event as a reference for the other one (*i.e.* it is based on several time references). It is then possible to express temporal properties between various references. In the synchronous domain it has proved to be adaptable to any level of description, from very flexible causal time descriptions to very precise scheduling descriptions [22]. Additionally, it is important to notice that the notion of multiform logical time is often used in everyday life. For instance, consider the sentence: "To be a world champion, a runner has to run 1200 meters and finish his run before the World Wide Record of 3min and 26s. The finish event corresponds to a distance and is put in a relation with the notion of time. The same idea is stated in *Req.3* and *Req.4* in Section II.

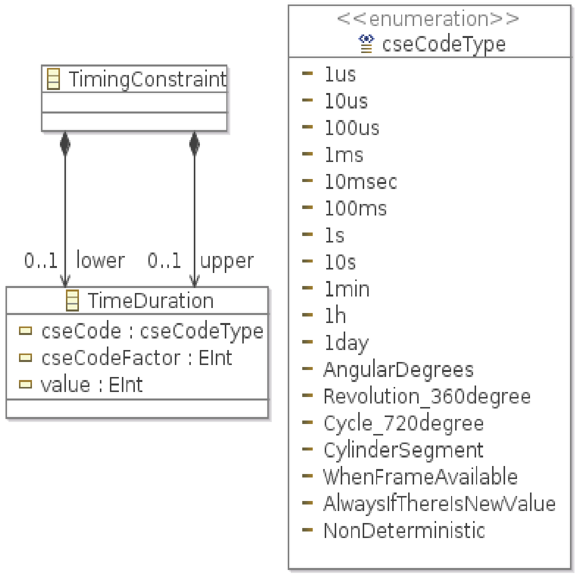

Figure 4. AUTOSAR Metamodel for Units

Such model of time has been precisely defined in the UML MARTE TIME profile [23]. However, in the context of the TIMMO-2-USE project, two main issues have been highlighted by the industrial automotive partners. First, the MARTE TIME profile is not aligned with either EAST-ADL or AUTOSAR. Second, there is a gap between the modeling of temporal properties in MARTE [24][25] and the already existing TADL leading to real changes in the habits of automotive engineers. Our proposal is then to extend TADL, while keeping the alignment with EAST-ADL and AUTOSAR, to enable the description of temporal properties by exploiting multiform and logical time.

IV. EXTENSION OF TADL2 WITH SYMBOLIC TIMING EXPRESSION

In this section we introduce the notion of timing specification. The goal of this specification is to provide Symbolic Timing Expression (STE). A STE is a way to specify parameterized expressions between different timebases as motivated in the previous sections. The description of the underlying concepts is done by describing the metamodel, splitted in several diagrams. The semantics of the concepts is given in natural language. In the metamodel, the *TimeBase* together with the *Dimension*, *Unit* and *TimeBaseRelation* address the requirements Req.3 & Req.4 and the multi timing aspect explained in Section II. The *TimingExpression* provides free variables, constants, values and operators to cover the need expressed in Req.2 for symbolic parameterized timing expressions.

In Section IV.A, we give the part of the TADL2 metamodel for multi time base extension. Section IV.B depicts the TADL2 extension in order to relate time bases to each other. In Section IV.C, we introduce the timing expressions in TADL2 which are *symbolic timing expression*, *variable timing expression* and *value timing expression*. Timing expressions are used by the timing constraints for EAST-ADL2 and AUTOSAR in order to express the duration such as maximum/minimum delay, period, jitter and tolerance duration. Section IV.D gives the alignment of TADL2 and AUTOSAR/EAST-ADL2 metamodels for the use of timing expressions in timing constraints.

A. A multi Time base extension for TADL2

Metamodel view

A *TimingSpecification* contains *TimeBase* which represents a discrete and totally ordered set of instants. An instant can be seen as an event occurrence called a "tick". It may represent any repetitive event in a system. Events may refer even to "classical" time dimension or to some evolution of a hardware part (rotation of crankshaft, distance …). The type of a *TimeBase* is a *Dimension*. The *Dimension* has a *kind* that represents the nature of the *TimeBase*. Of course, *Time*, *Angle* and *Distance* which are often used in the automotive domain are proposed. Additionally, *Logical* can be used to define a logical time reference. In particular, *logical* can be used to describe the units mentioned in the AUTOSAR metamodel (i.e. *WhenFrameAvailable*, *AnalysisIfThereIsNewValue* in Figure 4). Finally, *other* can be used for very specific applications.

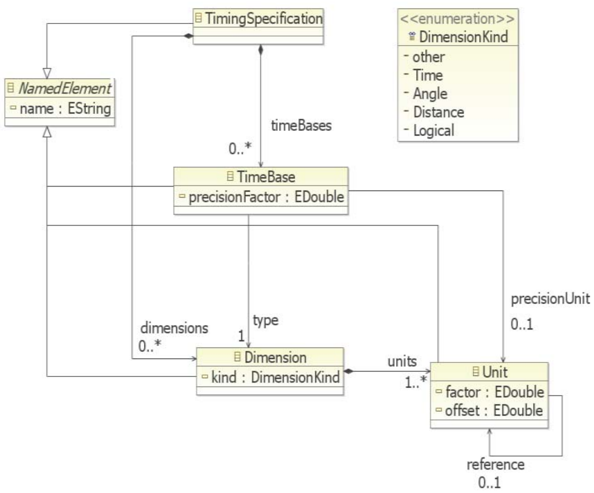

Figure 5. Part of the TADL2 Metamodel for Time Base and Dimension

A *Dimension* defines the set of units that can be used to express duration measured on a given TimeBase. The Dimension can be seen as the type of a TimeBase. Each *Unit* relates to another unit to enable conversions. The *factor*, *offset* and the *reference* attributes in the *Unit* are used for such conversions. Only linear conversions between units of the same dimension are allowed. As a unit conversion example, the unit *second = 1000 *millisecond* so *factor = 1000 and offset =0*.

Because a *Timebase* is a discrete set of instants, a discretization step is specified with the *precisionFactor* attribute which relies on a *precisionUnit*.

User view

We provide a textual concrete syntax for TADL2. Listing 1 shows a TADL2 *TimingSpecification* where two *Dimensions* are declared (*angle* and *physicalTime*). For each dimension, a list of units and attributes for their conversion expression are given.

```
1  TimingSpecification ts1 {
2
3    Dimension angle {
4      units {
5        degree{factor 1.0 offset 0.0},
6        revolution{factor 360.0 offset 0.0 reference degree}
7      }
8      kind  Angle
9    }
10
11   Dimension physicalTime {
12     units {
13       micros{factor 1.0 offset 0.0},
14       ms{factor 1000.0 offset 0.0 reference micros}
15       second{factor 1000000.0 offset 0.0 reference micros}
16     }
17     kind Time
18   }
19
20   TimeBase crk_angle {
21     dimension angle
22     precisionFactor 1.0
23     precisionUnit degree
24   }
25
26   TimeBase chrono_time {
27     dimension physicalTime
28     precisionFactor 0.1
29     precisionUnit micros
30   }
31
32   TimeBase universal_time {
33     dimension physicalTime
34     precisionFactor 0.1
35     precisionUnit micros
36   }
37
38   TimeBase ecu1 {
39     dimension physicalTime
40     precisionFactor 0.1
41     precisionUnit micros
42   }
43
44 }
```

Listing 1. Example of Dimension and TimeBase

The *angle* dimension has two units named *degree* and *revolution* where 1 *revolution* unit is equal to 360 *degree* unit (see lines 5 and 6). There is also a conversion for *micros*, *millisecond* (*ms*) and *second* in the *physicalTime* dimension (see lines 13 - 15).

Based on these dimension types, the *TimingSpecification* declares four *Timebases.* The *crk_angle* resp. *chrono_time* timebase is declared with a type *angle* resp. *physicalTime* (see lines 20 - 30).

The other two timebases are *universal_time* and *ecu1* (see Section II). To avoid the duplication in Listing 1 we did not show timebase declarations for *ecu2*, *ecu3*, *ecu4* and *ecu5* (see ECUs in Figure 3). For all *timebases*, a *precisionFactor* and a *precisionUnit* are given. For the *crk_angle* timebase, the *precision* means that this timebase is able to specify value with a precision of 1 angle degree.

The *Dimension* declaration is easily extendable to add new units but some of them (time, angle) are frequently used in a system. The objective would be to develop once and for all the *Dimension* definition including the different units. This can be done through dedicated libraries.

B. *TimeBase Relation in TADL2*

In the previous section, we introduce a *TimingSpecification* with different timebases. As stated by *Req.4* in section II, expressing relation between timebases is mandatory to build a global perception of time. AUTOSAR and EAST-ADL timing constraints can be synchronous (periodicity, etc.) or asynchronous (precedence, etc.). When these timing constraints refer to multiple TimeBases, it results in a partially ordered set of instants from these timebases and corresponds to the global temporal perception of system behavior.

Metamodel view

The *TimeBaseRelation* (see Figure 6) is used to give equivalence between different Timebases. More precisely, it specifies equality between a *left* and a *right* *TimingExpression* (see Section IV.C for the timing expression description).

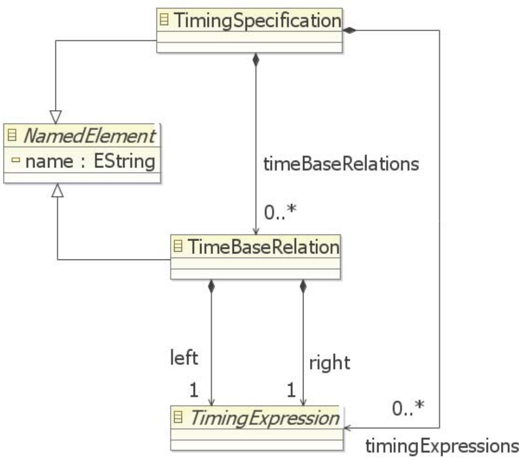

Figure 6. Part of the TADL2 Metamodel for TimeBase Relations

The goal is to allow expressing the same constraint on different TimeBases, if the TimeBases are related to each other.

User view

Listing 2 shows the TimeBaseRelations for the BBW example. As stated in *TC.5* in Section II, *ecu5* has a drift of 0.02 millisecond for each second compared to the universal time. Also, the *ecu5* TimeBase goes 2 times faster than TimeBases of *ecu1* to *4* (see *TC.6* in Section II).

```
1  TimingSpecification ts1 {
2
3    TimeBaseRelation tbr1 {
4        (1.0 degree on crk_angle) = (27.0 micros on chrono_time)
5    }
6
7    TimeBaseRelation tbr2 {
8        (1.0 second on universal_time) = (1.00002 second on ecu5)
9    }
10
11   TimeBaseRelation tbr3 {
12       (1.0 ms on ecu1) = (2.0 ms on ecu5)
13   }
14
15 }
```

Listing 2. Example of TimeBase Relation

The timing constraints *TC.5* and *TC.6* become the timebase relations *tbr2* and *tbr3* in Listing 2. To avoid the duplication, we did not show the timebase relations between *ecu2&ecu5*, *ecu3&ecu5* and *ecu4&ecu5*. They are the same as the timebase relation *tbr3*.

In addition to timing constraints related to the BBW example, in Listing 2 we added a timebase relation between timebases of different dimensions. *tbr1* states that 1 *degree* unit on *crk_angle* is equal to 27 *micros* unit on *chrono_time* (see lines 3-5).

C. *Timing Expression in TADL2*

The *TimingExpression* stands for all terms that denote time values in TADL2 and allows complex parameterized timing expression referring to multi timebases.

Metamodel view

There are three different timing expressions: *ValueTimingExpression*, *VariableTimingExpression* and *SymbolicTimingExpression*. Figure 7 shows the completeTADL2 metamodel including timing expression.

A *ValueTimingExpression* may have a unit and a time base as type. TADL2 is aimed to be a declarative language. Therefore, we have only free variables, constants and values. The *VariableTimingExpression* stands for free variables and constants. If a value is assigned to a variable, then the variable becomes a constant.

In *SymbolicTimingExpression*, the language integrates basic arithmetic and relation operators such as *addition*, *subtraction*, *multiplication*, *greater than*, and *less than* associated with timing values.

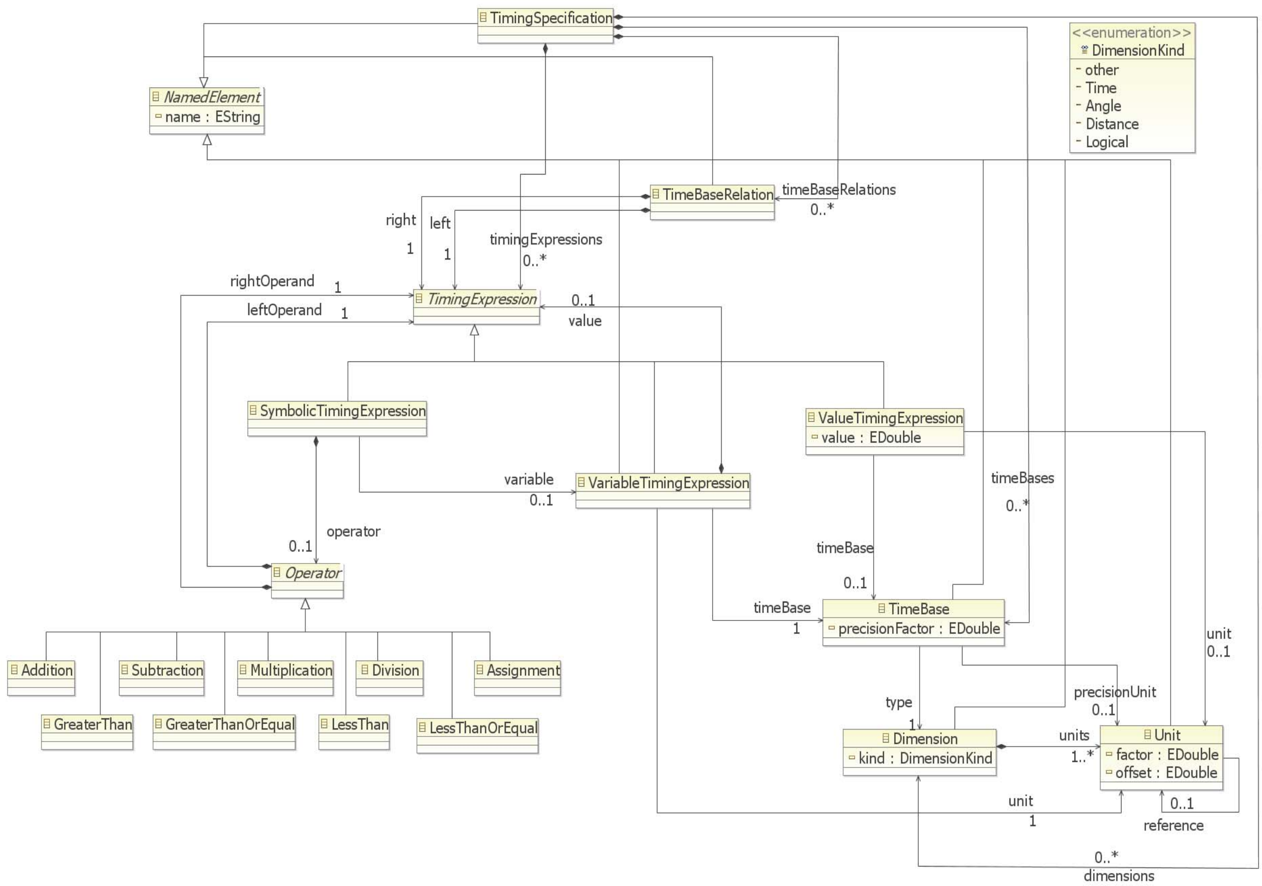

Figure 7. TADL2 Metamodel for Timing Specification

There are some implicit constraints in the TADL2 metamodel which are not shown in Figure 7. These constraints can be written in OCL form in order to check them in the metamodel. These constraints are in the following:

- The *SymbolicTimingExpression* cannot have both an *Operator* and a reference to the *VariableTimingExpression* (the association *variable* in Figure 7). It is not allowed to have an expression like {((X + Y) < Z)}.
- The left hand side of the *TimeBaseRelation* cannot be a *SymbolicTimingExpression* with an *Operator*. It can only be a *VariableTimingExpression* or a *ValueTimingExpression* with a *Unit* and a *TimeBase*.
- The right hand side of the *TimeBaseRelation* cannot be a *SymbolicTimingExpression* with a relation operator such as *Assignment* and *LessThan*. In the right hand side, it is possible to have a *SymbolicTimingExpression* containing an arithmetic operator with value or variable timing expressions with the same *Unit* and *TimeBases*. For instances, the following time base relations are not allowed: {((X + Y) = (5 ms on universal_time)} and {(1 degree on crk_angle) = (X < (5 ms on universal_time))}. On the other hand, it is possible to have a time base relation like in the following: {(1 degree on crk_angle) = (X + (5 ms on universal_time) }. Please note that the *Unit* should be *ms* and the *TimeBase* should be *universal_time* for X.
- Arithmetic operators cannot have right/left operands which are *SymbolicTimingExpressions* containing any relation operator. For instance, the following symbolic timing expression is not a valid timing expression: {((X < (5 ms on universal_time)) + Y)}.

Please note that the *TimeBaseRelation* is an equality for two different time bases written as timing expressions. It is different than the relation operator *Assignment*. Since we have only free variables and constants, the *Assignment* operator can be used only once for a variable in the left operand. The variable becomes a constant.

User view

Listing 3 extends the timing specification *ts1* with examples of timing expressions.

```
1  TimingSpecification ts1 {
2
3    TimeBaseRelation tbr4 {
4     (1.0 degree on crk_angle) = (var speed ms on universal_time)
5    }
6
7    { (speed := 36) }      // Symbolic Timing Expression (STE)
8
9    var X_VL ms on universal_time    // variable timing expression
10   { (X_VL < (200 ms on universal_time)) }      // STE
11   { (X_VL > (180 ms on universal_time)) }      // STE
12
13   var Y_AL ms on universal_time    // variable timing expression
14   { (Y_AL := 0.60* X_VL) }                   // STE
15
16   var Z_AL ms on universal_time   // variable timing expression
17   { (Z_AL := 0.40* X_VL) }                   // STE
18
19   { (Y_AL + Z_AL ≤ X_VL) }                  // STE
20
21   var Y_DL ms on universal_time  // variable timing expression
22   { (Y_DL := Y_AL - (20 ms on universal_time)) }      // STE
23
24    var Z_DL ms on universal_time  // variable timing expression
25    { (Z_DL := Z_AL - (10 ms on universal_time)) }      // STE
26
27    var FLWSacq ms on ecu1 := 20
28    var BPacq ms on ecu5 := 15
29    var Brake ms on ecu1 := 10
30
31  } // end of timing specification
```

Listing 3. Example Timing Specification with Timing Expressions

In Listing 3, we skip the dimension, timebase and timebase relation declarations already given in Listing 1 and Listing 2. Variables, constants and values given in Listing 3 conform to the ones expressed for time budgeting in Figure 2.

The *var* keyword is used for defining both free variables and constants. *speed, FLWSacq, BPacq* and *Brake* are defined as constants in lines 4, 27, 28 and 29.

Free variables are useful for characterizing parameters or variant in timing expression or when referring to already existing timing expression. Line 4 gives an example of the constant *speed* declared and accessed in the time base relation *tbr4*. The timebase relation *tbr4* is formed of one value timing expression (left hand side) and one variable timing expression (right hand side) (see lines 3-5). The value is given to the *speed* in line 7. The assignment of the value is a Symbolic Timing Expression (*STE*).

STE allows the assignment of intervals to variables. The variable X_{VL} comes from the timing constraint *TC.1* in Section II. X_{VL} is defined in line 9 with a value interval which comprises between 180 and 200 ms on universal time. Please note that different time bases can be used in upper and lower bounds of the value interval. In this case, the timebase relations are used to calculate the time interval for a single time base. *FLWSacq*, *BPacq* and *Brake* used for allocation of functions to ECUs are expressed as constants in Listing 3. The scope of all free variables and constants is the *ts1* timing specification.

Timing expressions can be used at different levels of abstraction in a design. In the next section we show how we can integrate timing expressions in TADL2 with EAST-ADL and AUTOSAR.

D. Integration of TADL2 with EAST-ADL/AUTOSAR

Metamodel view

In a EAST-ADL and AUTOSAR design, the *event* concept is the link between the timing constraints and the structural part of the system modeled. Figure 8 shows the representation of *Event* and *EventChain* in EAST-ADL and AUTOSAR.

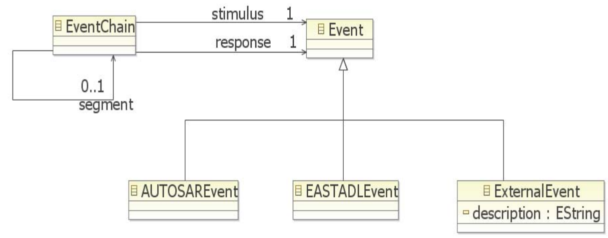

Figure 8. Events for EAST-ADL and AUTOSAR

The *Event* stands for all forms of the identifiable state changes that are possible to constrain with respect to timing. The *EventChain* is a container for a pair of events that must be causally related. The *EASTADLEvent* is an extension of the *Event* and it embodies an event of the form defined by EAST-ADL (see also the *AUTOSAREvent* in Figure 8 for AUTOSAR). There is another extension of the *Event* named *ExternalEvent* for some particular form of state change in the external physical world. TADL2 provides timing expressions used by timing constraints. Figure 9 illustrates the integration of the TADL2 metamodel with the EAST-ADL metamodel for one of the EAST_ADL timing constraints.

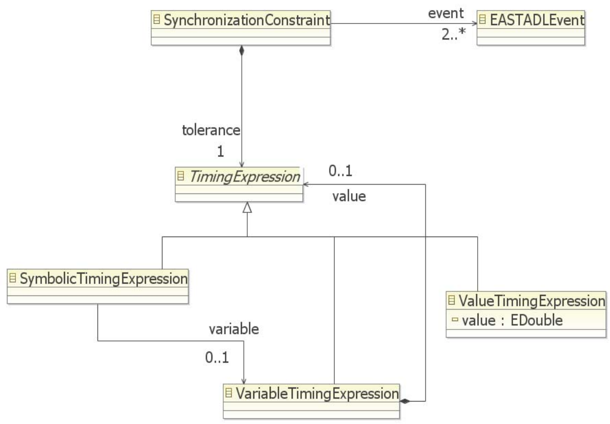

Figure 9. Integration of TADL2 with EAST-ADL

In Figure 9, we show how the *EASTADLEvent* is linked to the *TimingExpression* with the *SynchronizationConstraint* which describes how tightly the occurrences of a group of events follow each other. For AUTOSAR integration, the *EASTADLEvent* in Figure 9 is replaced with the *AUTOSAREvent*.

User view

Listing 4 gives an example synchronization constraint in EAST-ADL with textual concrete syntax (see the timing constraint *TC.3* in Section II).

```
1   Event firstWheelBrakeActuation { }
2   Event secondWheelBrakeActuation { }
3   Event thirdWheelBrakeActuation { }
4   Event fourthWheelBrakeActuation { }
5
6   SynchronizationConstraint sc1 {
7     events firstWheelBrakeActuation,
8            secondWheelBrakeActuation,
9            thirdWheelBrakeActuation,
10           fourthWheelBrakeActuation
11
12    tolerance = (5 ms on universal_time)
13
14  }
```

Listing 4. Example Synchronization Constraint in EAST-ADL

The constraint is about the maximum tolerated time difference between the first and last wheel brake actuation. The brake actuation is defined for each wheel as an event (see lines 1-4). For these events, the synchronization constraint *sc1* has the attribute *tolerance* which is a *ValueTimingExpression* (see line 12).

V. Modeling Environment for TADL2

We have a TADL2 editor that supports textual concrete syntax. The TADL2 metamodel is implemented with *ecore* [27] in Eclipse Modeling Framework (EMF). The textual concrete syntax for TADL2 is generated by using *Xtext* [28] which is a framework/tool for development of external textual domain specific languages.

VI. Conclusion

This paper presents new features of the Time Augmented Description Language for the modeling of multiple temporal referential (TimeBase) in automotive design (clock from different ECU, motor position, etc.). The relation between these timebases and the possibility to have a suitable set of arithmetic operators mixing symbolic identifiers and referring to different timebases are also parts of the extension. The new concepts of TimeBase, TimeBase Relation, Timing expressions and their associated semantics are presented and illustrated in a BBW example. An alignment of TADL2 and AUTOSAR/EAST-ADL2 metamodels is proposed in order to benefit from these extensions in AUTOSAR and EAST-ADL2.

TADL2 supports a high level modeling of timing aspects of a system and a refinement of timing constraints through multiple levels of design. We progress henceforth on the way of analyzing timing constraints by using model transformation techniques to go towards simulation and analysis tools. One potential candidate for simulation is the Timesquare environment [29] and the associated language CCSL [24] which allow multi clocks system specification. In a second step and for a formal analysis of TADL specifications, a synchronous language environment such as SCADE [30] could be envisaged.

Acknowledgment

This document is based on the TIMMO-2-USE project in the framework of the ITEA2, EUREKA cluster N°3674. The work has been funded by The French Ministry for Industry and Finances, the German Ministry for Education and Research (BMBF) under the funding ID 01IS10034, and the Swedish governmental agency for innovation systems (VINNOVA). The responsibility for the content rests with the authors.

References

[1] Tindell, K.W.; Hansson, H.; Wellings, A.J. Analysing real-time communications: controller area network (CAN). Real-Time Systems Symposium, 1994., Proceedings. Dec 2004, **ISBN:** 0-8186-6600-5

[2] Berwanger, J., C. Ebner, et al. (2001). FlexRay--The Communication System for Advanced Automotive Control Systems. SAE World Congress, Detroit, SAE Press paper 2001001-0676.

[3] AUTOSAR AUTomotive Open System Architecture. *http://www.autosar.org*

[4] The East-EEA Project. Defnition of language for automotive embedded electronic architecture, ITEA, 2004.

[5] AUTOSAR Specification of Timing Extensions, 1.1.0, AUTOSAR Release 4.0.2, 2010-11-03, AUTOSAR Development Cooperation.

[6] The ATESST Consortium. ATESST - the Modelling Approach: Overview of the East-ADL2. ITEA, Tech. Rep., 2007, deliverable D.3.1. [Online]. Available: http://www.atesst.org

[7] N. Feiertag, K. Richter, J. Nordlander, J. Jonsson. A Compositional Framework for End-to-End Path Delay Calculation of Automotive Systems under Different Path Semantics. IEEE Real-Time System Symposium (RTSS), Workshop on Compositional Theory and Technology for Real-Time Embedded Systems (CRTS'08). Barcelona, Spain, December 2008

[8] The ITEA TIMMO Project. http://timmo-2-use.org/

[9] O. SEI, "OSATE An extensible source AADL tool environment," SEI AADL Team technical Report, 2004.

[10] Systems Modeling Language (SysML) Specification 1.1. OMG Document Number: ptc/08-05-17. http://www.sysml.org

[11] P. Cuenot, P. Frey, R. Johansson, H. Lönn, M-O Reiser, D. Servat, R. Tavakoli Koligari, D.J. Chen. Developing Automotive Products using the EAST-ADL2, and Autosar Compliant Architecture Description Language. European Congress on Embedded Real-Time Software (ERTS). Toulouse, France. January 2008

[12] H. Espinoza, H. Dubois, S. Gerard, J. Medina, D.C. Petriu. Annotating UML Models with Non-Functional Properties for Quantitative Analysis. Proc of MODELS'2005 Satellite Events, Lecture Notes in Computer Science, Springer, 2006

[13] J. Stankovic. VEST: A Toolset for Constructing and Analyzing Component based Operating Systems for Embedded and Real-time Systems. University of Virginia TR CS-2000-19, July 2000.

[14] F. Singhoff, J. Legrand, L. Nana, and L. Marce. Scheduling and Memory Requirements Analysis with AADL. Proceedings of the International ACM SIGAda Conference, Atlanta, USA, November 2005.

[15] H. Blom, R. Johansson, and H. Lonn. Annotation with Timing Constraints in the Context of EAST-ADL2 and AUTOSAR–the Timing Augmented Description Language.

[16] A. Hamann, R. Henia, M. Jersak, R. Racu, K. Richter, and R. Ernst. Symta/s-Symbolic Timing Analysis for Systems. In WIP Proc. Euromicro Conference on Real-Time Systems 2004 (ECRTS'04), pp. 17–20.

[17] K. Klobedanz, C. Kuznik, A. Thuy, and W. Mueller. Timing Modeling and Analysis for AUTOSAR-based Software Development - a Case Study. In Design, Automation & Test in Europe Conference & Exhibition (DATE), 2010. IEEE, 2010, pp. 642–645.

[18] L. Lamport. Time, Clocks, and the Ordering of Events in a Distributed System, Comm. of the ACM, (July 1978)

[19] C. J. Fidge. Logical Time in Distributed Systems. Comput., vol. 24, no. 8, pp. 28-33, Aug. 1991.

[20] G. Berry. The foundations of Esterel. In Proof Language and Interaction: Essays in Honor of Robin Milner. Cambridge, MA: MIT Press, 2000.

[21] A. Benveniste, P. Guernic, and C. Jacquemot. Synchronous Programming with Events and Relations: The Signal Language and its Semantics. Sci. Comput. Program., vol. 16, pp. 103–149, 1991.

[22] F. Boussinot, R. De Simone. The ESTEREL Language. Proceedings of the IEEE, 79(9):1293–1304, Sept. 1991.

[23] The ProMARTE Consortium, UML Profile for MARTE, beta 3, Object Management Group, May 2009, oMG document number: ptc/2009-05-13.

[24] F. Mallet, C. André, and R. de Simone. CCSL: Specifying Clock Constraints with UML/Marte. ISSE, 4(3):309–314, 2008.

[25] C. André, F. Mallet, and R. de Simone. Modeling time(s). In G. Engels, B. Opdyke, D. C. Schmidt, and F. Weil, editors, MoDELS, volume 4735 of Lecture Notes in Computer Science, pages 559–573. Springer, 2007.

[26] EAST-ADL Language Specification, , http://www.atesst.org/home/liblocal/docs/ATESST2_D4.1.1_EAST-ADL2-Specification_2010-06-02.pdf

[27] Eclipse Modeling Framework (EMF). http://www.eclipse.org/modeling/emf/

[28] XText. http://www.eclipse.org/Xtext/

[29] J. DeAntoni, F. Mallet, and C. André, "TimeSquare: on the formal execution of UML and DSL models," Tool session of the 4th Model driven development for distributed real time systems, 2008.

[30] Parosh Aziz Abdulla, Johann Deneux, et al. "Designing Safe, Reliable Systems using *Scade* " In Proc. ISoLA 2004 , 2004