Towards Formal Energy and Time Aware Behaviors in EAST-ADL: An MDE Approach

Eun-Young Kang, Gilles Perrouin, Pierre-Yves Schobbens
PReCISE Research Centre
University of Namur, Belgium
{eykang,gilles.perrouin,pyschobb}@fundp.ac.be

Abstract—Energy-aware real-time (ERT) systems are increasingly complex and have pervaded various areas, from automotive to telecommunication systems. Dedicated UML-based modeling languages, such as EAST-ADL or MARTE have been proposed to harness this complexity. However, they provide limited support for modeling ERT constraints, in particular continuous energy consumption. To cope with this issue we introduce a formal interchange language, eXtended Function-block Graphs (XFG), for modeling and analysis of ERT behaviors. An XFG UML profile augmenting EAST-ADL and MARTE is presented to facilitate modeling those behaviors by means of state machines. A set of mapping rules is proposed to automatically transform such profiled models into the XFG language.

I. INTRODUCTION

Nowadays most industrial domains (automotive, railways, aeronautics, etc.) develop systems that are complex, multidisciplinary, critical, real-time, and resource-constrained. Model-Driven Engineering (MDE) is increasingly adopted for the development of such systems, where models are analyzed to establish their properties at early phases, and refined to concrete and executable system software. The use of formal methods is the cornerstone to ensure model correctness, and preservation of properties through refinements.

Various initiatives have addressed the definition and analysis of system engineering models as extensions (profiles) of the Unified Modeling Language (UML) [17] such as SysML [14], MARTE [15] and EAST-ADL [4], the latter two being the focus of this paper. MARTE is presented to cope with time and resource constrained aspects. EAST-ADL, which is based on functions (equivalent to blocks in SysML and composite structures in UML), provides support for architectural specification and timing constraints for automotive embedded systems. Nevertheless, MARTE and EAST-ADL aim to describe energy-aware real-time (ERT) behaviors of a system, they cannot be formally verified and analyzed due to the lack of formal semantics. Neither language provides support for modeling continuous energy consumption with different rates.

A system in EAST-ADL is described by Functional Architectures (FA) at different abstraction levels. The FA is composed of a number of interconnected FunctionPrototypes ($F_P$), where each prototype is an instantiation of FunctionType ($F_T$). Functions have ports for communication. A function may be a basic function, or have an internal structure, which consists of sub-functions connected by channels. These concepts have not yet received a formal semantics. We can divide the semantics into two parts: 1) intra-function semantics where the basic functions are described by separate notations, due to the specific needs of each discipline (mechanics, software, electronics, etc.). Typical examples are Simulink, SCADE, Modelica, etc. Nevertheless, there is a need for a common semantic denominator to ensure sound system-wide analysis; 2) inter-function semantics deal with parallelism, communication, hierarchy and compositionality.

In order to alleviate the aforementioned restrictions, we introduce eXtended Function-block Graphs (XFG), an interchange format for formal modeling and analysis of ERT systems. XFG aims to establish interoperability of various tools by means of model transformations to and from XFG. To formally specify the ERT behaviors at the UML level, we extend UML state machines with a UML profile, called XFG profile, that integrates relevant concepts from EAST-ADL and MARTE profiles. Such profiled models are automatically translated into analyzable XFG by model transformations. In this way, developers can use familiar notations, while benefiting from formal verification.

Section II introduces the XFG profile for modeling ERT behaviors in EAST-ADL using UML state machines. Formal syntax and semantics of the XFG language are defined in Section II. The transformation of UML models stereotyped with the XFG profile to XFG is discussed in Section IV. Section V presents related work. Conclusion and future work are presented in Section VI.

II. ENERGY-AWARE TIMED UML STATE MACHINES

In this section, we introduce our modeling approach to ERT systems. Our running example is a Brake-By-Wire (BBW) system. A part of the BBW units are depicted in Fig.1. EAST-ADL FA (shown on left side in Fig.1) is composed of two connected $F_P$s typed by the Pedal and BrakeController $F_T$s. To model $F_T$ attributes, we use MARTE’s Non-Functional Properties package elements, NFP_Energy and time, for defining energy and clock constraints. The overall execution time of $F_T$ is provided with either EAST-ADL ≪TimingConstraints≫ or as a part of MARTE’s ≪ResourceUsage≫ stereotype. Intra behaviors of $F_T$ are modeled by means of UML state machines (middle of Fig.1). We define a UML profile extending state
machines, XFG profile, to model real-time, continuous energy consumption and urgency semantics. Our state machines are “flat” (no orthogonal regions are required), i.e., we do not deal with the internal parallelism of state machines, since parallelism is provided in the structural part of EAST-ADL and the parallel behavior is modeled by interacting functions.

As a general rule, invariants (timing and energy constraints), guards, and effects on transitions are expressed as XFG profiles according to the XFG language introduced in Section III (also refer to [9] for a complete definition): 1) Each state of a state machine is associated with a time constraint. This time constraint is modeled in terms of a clock-related XFG expression and attached to the state as its invariant. EAST-ADL time triggers can be interpreted by the engineer as time constraints on states; 2) For a state where the system may consume energy continuously, the XFG profile offers the $\langle$XFGContEnergy$\rangle$ stereotype. It allows the specification of a consumption rate ($\text{rate}$), reference to the clock ($\text{clock}$) and energy attribute concerned, and an expression ($\text{expr}$) linking these elements as depicted by the comment attached to getTorque state in Fig.1; 3) Transitions model action behaviors (“effect”) occurring between states, and are controlled by guards. Choices are used to form complex paths between states.

Mapping between the state machine and its corresponding $F_T$ ports is performed as follows: receiving data from input ports is modeled using ReceiveSignalEvent and sending data to an output port is modeled using SendSignalAction. For “data sending” we use UML activities allowing detailed behaviors (right side in Fig.1). Energy consumption on transitions is discrete and is modeled by the $\langle$XFGDiscEnergy$\rangle$ stereotype. Finally, urgent transitions (for which no time is spent in their source state) can be decorated via $\langle$XFGUrgent$\rangle$.

### III. Extended Function-block Graphs

State Diagrams with MARTE and EAST-ADL profiles are automatically translated into our ERT interchange format XFG (eXtended Function-block Graphs) that supports formal modeling and analysis of intra- and inter- behaviors of $F_T$s in EAST-ADL. An XFG system consists of a finite number of processes. The control part of any process is described as a finite state machine. Transitions of an XFG process can be marked as urgent, implying that they should be taken as soon as they are enabled. Processes of an XFG see their executions interleaved. They communicate by means of shared variables or by synchronous value passing. An XFG permits two-way synchronization communication on complementary input and output actions, as well as broadcast actions.

The XFG format is a clear text description based on Hybrid and Timed Automata (TA) [1] with language constructs to capture the axiomatic and operational specification of function aspects, and ERT behavior [9]. To define the dynamic semantics of XFG, we use some abstract syntax domains that are assumed to be provided by the data model:

**Definition 1** Given a finite set of variables $V$ and a subset of clock variables $V_c \subseteq V$, a data language provides the following syntactic domains:

- $\text{Expr}$: value expressions (over the set $V$ of variables),
- $\text{Bexpr} \subseteq \text{Expr}$: the subset of Boolean expressions.

We assume a universe $\text{Val}$ of values that includes the set $\mathbb{R}_{\geq 0}$ of non-negative real numbers and the Boolean values. A valuation is a mapping $\rho : V \mapsto \text{Val}$ from variables to values such that $\rho(c) \in \mathbb{R}_{\geq 0}$ for all $c \in V_c$. For a valuation $\rho$ and $\delta \in \mathbb{R}_{\geq 0}$ we write $\rho[+\delta]$ to denote the environment that increases each clock in $V_c$ by $\delta$:

$$
\rho[+\delta](v) = \begin{cases} 
\rho(v) + \delta & \text{if } v \in V_c \\
\rho(v) & \text{otherwise}
\end{cases}
$$

We define a core syntax for an XFG system, on which the dynamic semantics is based. An XFG is defined as a single, global transition system. At this level we assume that type correctness has been checked. For detailed syntax and semantics of the XFG language, we refer the reader to [9].
Definition 2 An XFG process is a tuple \( \langle Dtype, \text{Init}, L, l_0, I, E, U, CP \rangle \) where

- \( Dtype : V \rightarrow \{ \text{disc}, \text{cont}, \text{clock} \} \) assigns to each variable a dynamic type: discrete, continuous, or clock.
- The sets \( V_{\text{disc}}, V_{\text{cont}}, V_{\text{c}} \) are defined as \( V_t = \{ v \in V \mid Dtype(v) = t \} \) for \( t \in \{ \text{disc}, \text{cont}, \text{clock} \} \).
- \( \text{Init} \in V_{\text{disc}} \) indicates the initial condition. A set of dotted variables \( \bar{V} \in V_{\text{disc}} \) represents different rates of increasing energy.
- \( L \) is a finite set of locations,
- \( l_0 \in L \) is the initial location,
- \( I : L \rightarrow \text{Bexpr} \) assigns an invariant to each location,
- \( H \) is a finite set of synchronizing action labels,
- \( E \subseteq L \times \text{Bexpr} \times 2^V \times \text{Expr} \times (H \cup \tau) \times L \) is a set of transitions, represented as tuples \( \langle l, g, h, u, l' \rangle \) where \( l \in L \) is the source location, \( g \in \text{Bexpr} \) is the guard, \( h \in H \) is a synchronization label \( \{ h!x, h?v \mid \{ x \} \subseteq \text{Expr}, \{ v \} \subseteq V \} \), where \( x \) and \( v \) are either empty or sequences of expressions or variables, \( u \subseteq V \times \text{Expr} \) is an update, and \( l' \in L \) is the destination location.
- \( U \subseteq E \) identifies the subset of urgent transitions.
- \( CP : L \cup E \rightarrow \mathbb{R}^{\geq 0} \) assigns an energy consumption to each location and transition.

The semantics of an XFG is defined in terms of a timed structure.

Definition 3 The operational semantics of an XFG process is given as a timed transition system \( \langle S, s_0, T \rangle \) where

- \( S = \{ (l, \rho) \in L \times \rho[+\delta]|(v) \} \)
- \( s_0 = \{ (l_0, \rho_0) \} \)
- \( T \subseteq S \times (E \cup \mathbb{R}^{\geq 0}) \times S \) such that:
  - For any \( e = \{ (l, g, h, u, l') \} \in E \) and \( \{ (l, \rho), (l, \rho'[u]) \} \subseteq S : (l, \rho) \xrightarrow{e} (l', \rho'[u]) \)
  - For any \( \delta \geq 0 \) and any \( \{ (l, \rho), (l, \rho'[+\delta]) \} \subseteq S : (l, \rho) \xrightarrow{\delta} (l, \rho[+\delta]) \)
  - To each such transition step, we associate an energy consumption defined by

\[
\begin{align*}
CP((l, \rho) \xrightarrow{e} (l', \rho'[u])) &= CP(e) \\
CP((l, \rho) \xrightarrow{\delta} (l, \rho[+\delta])) &= CP(l) \cdot \rho[+\delta]
\end{align*}
\]

A run \( \pi \) of XFG is a finite of infinite sequence of steps with no time-stuttering. The energy consumption of \( \pi \) is the accumulated consumption of steps along the run. An XFG is a finite set of XFG processes. With any XFG process we associate a timed structure, allowing continuous consumption of energy, whose states are given by the active locations of the XFG process and the evaluations of the underlying variables.

IV. TRANSLATION

A. MTL

Transforming UML models into the XFG textual specifications amounts to defining Model-to-Text (M2T) transformations. We have selected Acceleo\(^\text{[4]}\) for the following reasons:

1) it is a free implementation of OMG’s MOF Model to Text Language (MTL); 2) it integrates easily with the Papryus UML tool used for EAST-ADL and MARTE modeling within the eclipse platform.

An MTL program consists of transformation rules (called templates), which are organized in modules. A template is formed both by an immutable text and by expressions enclosed by square brackets. When applied to an actual model, these expressions are substituted by the result of their evaluation. Navigation amongst model elements is performed using the Object Constraint Language (OCL) syntax.

We are currently implementing the transformation of XFG profile models into the XFG textual format. Code snippets can be found in the companion technical report \([9]\), in the following section we describe the main mappings realized by the transformation, focusing on state machine translation.

B. Mapping Rules

1) Mapping Control States: By the first rule (see R1 in Fig.2), the initial UML pseudostate and the state targeted by its unique transition are mapped to the initial location of an XFG process corresponding to \( F_T \) whereby the state machine is defined. This mapping rule is motivated by the fact that UML constrains the initial pseudostate (one unique transition, on which neither trigger nor guard can be added), which does not exist in XFG. There are two kinds of ending states: Final and Terminate (pseudo)states (R2 and R3). A final state stops all flows in state machines. It is mapped to the final location of the XFG process (denoted as End in Fig.2), which has an outgoing transition. This transition is associated with a synchronization channel sending out signals or events to the external XFG processes. A terminate state indicates that the state machine is not responding to any other events, which means that it is about to be destroyed. The XFG equivalent of a terminal state is an absorbing location, i.e., the location does not have any outgoing transitions.

Regular states are mapped to conventional locations in XFG. State invariants are simply translated into location invariants. Regarding continuous energy consumption, we retrieve the expression from the \(<\text{XFGContEnergy}>\) stereotype and map it to a location invariant. Choice pseudostates (R4) indicate a set of possible targets, at most one of which will be selected on the basis of a guarding condition. In XFG,
a choice can be replaced by the same number of transitions that are separated from each other with each one having its own guard. One transition will be executed at a time which simulates the behavior of the choice.

2) Mapping Flows: UML transitions are mapped to XFG transitions. For guards, discrete energy and effects expressions on the transitions are mapped to XFG guards, energy and updates. \(<\text{XFGUrgent} >\) is mapped straightforwardly to urgent transitions (R5).

3) Mapping communication between \(F_T\): To deal with interactions between \(F_T\), we refer structural EAST-ADL constructs, i.e., \(<\text{FunctionConnector}>\) (FC) between \(F_T\), and they are translated to synchronization or broadcasting channels between XFG processes: The name of a channel is identical to the name of the corresponding FC. This FC is associated with the output port of \(F_{1_T}\) and the input port of \(F_{2_T}\). Data is passed through the channel (which mirrors the FC) and is mapped from the UML attribute owned by \text{SendSignalAction} and \text{ReceiveSignalEvent}.

V. RELATED WORK

Several works have been carried out on the translation from UML models to analyzable TA models [3, 7, 12]. These approaches did not address how to transform UML diagrams annotated with MARTE stereotypes into TA models, especially for the formal modeling of resource-aware systems. Recently Andrade et al. [2] proposed a method for mapping SysML activity diagram annotated with MARTE into Petri nets [13]. However, only events’ flows were dealt with, data related issues were not considered. Yang et al. [20] investigated the data related issue in detail using some extensions of Petri nets, which, however, did not support a hybrid variable in particular regarding different energy consumption rates during execution. Feng et al. [6] translated EAST-ADL functional models written in activity diagrams into the PROMELA [16], and in contrast to our work, it did not allow the integration of timing constraints. In our early work [10], [11] an integration effort towards formal modeling and analysis of EAST-ADL models based on timing constraints was investigated where the models were manually translated into TA formats in UPPAAL series tools [5, 19] without addressing energy-aware behaviors.

VI. CONCLUSION

A formal interchange format XFG expressed in structured operational semantics for modeling and analysis of ERT systems is introduced. It provides a sound basis for interdisciplinary (intra- and inter-block) semantics of EAST-ADL. A UML state machine is used as intra-block visual language, and its extended profile, XFG profile, is defined in order to deal with real-time, continuous energy consumption and urgency semantics. This XFG profiled UML model is automatically transformed into the analyzable XFG textual language, which can be used for relevant verification tools. In consideration of inter-block semantics, structural constructs \(<\text{FunctionConnector}>\) in EAST-ADL are translated into the corresponding synchronization channels in XFG. A set of mapping rules for the transformation is proposed.

As continuous work, we will automatically map XFG to UPPAAAL CORA [18] or HYTECH [8] and translate the constraints in XFG profiles as properties to be validated by these tools. Furthermore, we will study the semantic equivalence between UML models and XFG. We will also investigate a direct extension of EAST-ADL that can support energy- and time aware behavior constructs by refining the native behavior and time support in EAST-ADL.

ACKNOWLEDGMENT

This work is funded by FNRS. We wish to acknowledge Henrik Lönn (Volvo Technology Corp., Sweden) for his valuable feedback.

REFERENCES


