

Managing Functional Safety Processes for Automotive E/E Architectures in Integrated Model-Based Development Environments

Nico Adler, Stefan Otten, and Melanie Schwär
FZI Research Center for Information Tech.

Klaus D. Müller-Glaser
Karlsruhe Institute of Technology

ABSTRACT

The international standard ISO 26262 for functional safety of road vehicles claims processes and requirements for the entire product lifecycle of automotive electric and electronic systems. The demanded activities and work products within the standard are highly interconnected. Additionally, references to exemplarily external quality management standards or commonly recognized industry sources are given. Therefore, the application of functional safety processes in distributed development is challenging regarding description, understanding, analysis and planning of processes. To overcome these inconveniences, we provide a meta model extension for model-based architecture description languages regarding process description, organizational structures and resource assignment. This is related to the established "Business Process Model and Notation" (BPMN) according to ISO/IEC 19510:2013. Our integration in a domain specific environment for large-scaled electric and electronic (E/E) architectures provides graphical modeling of processes. The process model artifacts can be linked to data model artifacts of the E/E architecture. Additionally, analysis and planning based on the process model is supported. As a research case study this was applied to the functional safety processes described in ISO 26262.

CITATION: Adler, N., Otten, S., Schwär, M., and Müller-Glaser, K., "Managing Functional Safety Processes for Automotive E/E Architectures in Integrated Model-Based Development Environments," *SAE Int. J. Passeng. Cars – Electron. Electr. Syst.* 7(1):2014, doi:10.4271/2014-01-0208.

1. INTRODUCTION

Process modeling becomes increasingly important in particular for the product development of safety-critical systems in the automotive, railway, avionic and medical device domain, due to high system complexity and large supply chains. Additionally, besides ensuring high product quality, legal requirements as well as national and international standards must be adhered to.

Focusing on functional safety in the automotive domain, the standard ISO 26262 [1] is normative and already applied to industry projects. Considering the functional safety aspects of automotive systems more closely, it becomes obvious that the importance of process modeling is growing due to novel features for exemplarily driver assistance or electrically powered steering. Consequently, the systems are expanding in complexity. The impact on safety and the demands regarding reliability are increasing. Additionally, system safety has to be linked with security, which is an upcoming challenge. To make this complex topic manageable especially for non-safety experts, integrated process modeling is one of the most appropriate methods.

In [2] the process model represents an "abstract description of an actual or proposed process". The single elements in the process can be performed by a human or machine. Most common perspectives for process representation according to [2] are functional, behavioral, organizational and informational. Requirements for process modeling languages in context of software engineering are given in the work of [3]. Semantic richness, conformity to the UML standard, graphical representations and support of multiple views, executability, modularity, and formality are in focus of this work. Based on these requirements, UML-based software process modeling languages are compared and discussed in terms of the current practice in software engineering. The work of Becker [5] describes that process modeling becomes more important for other purposes besides software engineering. Therefore, general techniques are presented for adjusting models in terms of specific purposes.

Process models give an overview of the process flow and allow assignment of resources. Overall, this facilitates the understanding of functional safety processes and forms the basis for subsequent analysis and planning of processes. Additionally, it supports job training during time-intensive daily

business, saving and internal propagation of intellectual property, integration of project experiences, counteracting the lack of transparency, clear doing of business, saving expert knowledge and methodology skills [4].

Process modeling in context of ISO 26262 is topic in the work of [6] [7]. Requirements of ISO 26262 were translated into concrete activities and roles were assigned. To provide the information together with the inputs and outputs of the activities, a framework for a spreadsheet is described. Additionally the “Software & Systems Process Engineering Metamodel” (SPEM) [8] and Eclipse Process Framework (EPF) [9] Composer were used to provide a process framework for safety-related projects.

In the work of [10] the benefit of formal methods and languages for business process management is mentioned. Complementary to this, in the work of [11] different process model notations are described. Interesting are the Petri Nets based on the work of Petri [12] and the “Business Process Model and Notation” (BPMN) [13]. Petri Nets provide an intuitive and simple modeling with their bipartite graphs, furthermore the models can be used for analysis [11]. However, Petri Nets have difficulties to capture data and time-related aspects [11].

The “Business Process Model and Notation” (BPMN), actual version 2.0.1 [14], was standardized by the Object Management Group (OMG) and formed the basis for the ISO/IEC 19510:2013 [13], published in July 2013. This shows the significance of BPMN. Tools, such as ARIS [15] or Enterprise Architect [16], support modeling valid to the BPMN specification. The BPMN 2.0 Modeler for Eclipse [17] also supports the graphical modeling of BPMN diagrams.

If process modeling is integrated in a domain-specific environment, process analysis and planning can be facilitated based on the models. In addition, the E/E architecture data model and the process model can be linked and an overview of the overall process is ensured. In this paper we describe a concept for process modeling, especially regarding the development of complex electric and electronic architectures. The concept focuses on the usage of basic constructs provided by BPMN as this language is commonly recognized and extendable. Additionally, a prototype implementation is presented as well as an exemplary application to the automotive functional safety standard ISO 26262.

The paper is structured as following: In the following [Section 2](#) the concept for process modeling in an integrated model-based environment is described. [Section 3](#) provides the proposed meta model extension regarding process flow and organizational structures. In [Section 4](#), the process analysis and planning is presented based on the proposed extensions. [Section 5](#) describes our research prototype implementation in a model-based tool environment for large-scaled electric and electronic architectures. In [Section 6](#) the concept is applied to

the functional safety standard ISO 26262. [Section 7](#) gives a discussion of the results. [Section 8](#) summarizes the paper and gives an outlook to future work.

2. CONCEPT FOR INTEGRATED PROCESS MODELING AND MANAGEMENT

During concept and development phase of complex systems, different appropriate abstraction layers and perspectives are relevant for the model-based description of E/E architectures, cf. [18] [19]. Requirements can be described for specification of the complete system. Based on a logical architecture, the system can be refined into software and hardware architecture. Additionally, geometric parameters of the system can be described. Traceability across different abstraction layers must be ensured to facilitate consistency especially in context of safety-related systems.

For the approach of integrated process modeling, we propose a meta model for process description which is orthogonal to the exemplary presented abstraction layers, as shown in [Figure 1](#). This process model can be linked to the architecture description language to facilitate combination with the data model.

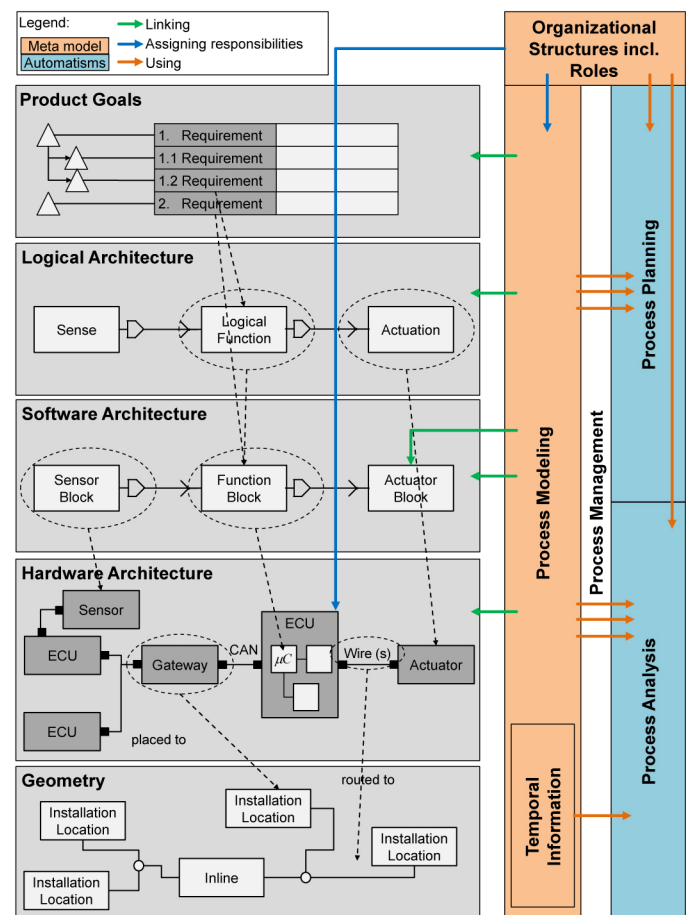


Figure 1. Concept for integrated process modeling and management in E/E architecture description languages.

BPMN provides relevant meta model constructs for the description and modeling of processes including resource allocation and data handling. Due to the challenge of combining process models with architecture models, meta model constructs of BPMN have to be partially adapted to be integrated in E/E modeling environments. For special purposes, exemplarily process analysis, the constructs additionally have to be extended. This can be performed concurrently in accordance with the extensibility of BPMN, described in [13] Part 7.7. The extensibility mechanism can be used to “satisfy a specific need, such as the unique requirements of a vertical domain, and still have valid BPMN Core”.

For the modeling of processes, BPMN provides three different basic types of sub-models: processes, choreographies and collaborations according to [13] Part 7.2.1. Whereas the process diagram depicts all activities and events within a workflow, the choreography defines the expected behavior between interacting participants. The collaboration diagram describes the interaction between at least two business entities, represented by pools. This approach focuses on modeling of workflows with usage of the basic constructs of the BPMN process model in terms of E/E architecture development.

A link of process artifacts to different artifacts in the abstraction layers of a model-based environment is facilitated based on the process meta model constructs. This extends existing abstraction layers within architecture description languages and facilitates the interaction and seamless architecture development. Additionally, the link supports the interchange of information.

For complex processes and persons involved, often not limited to single companies, different perspectives have to be provided to cope with the overall focus and partial interests. Also the granularity of the process model is of great significance and depends on the actual work in focus and internal application of the processes.

For the description of different organizational structures in context of the modeled process, a meta model is proposed. Different internal or external organizational structures and hierarchies as well as single roles can be modeled focusing on their interaction with the process. Therefore, constructs for assignment to specific process artifacts have to be provided. The assignment to the architecture data model in terms of responsibilities for certain data artifacts is also in focus.

Based on the extended process meta model, the meta model for organizational structures and their links to the E/E data model, different methods for analysis of the process can be provided. Regarding different perspectives, exemplarily the process owner, these analyses facilitate an evaluation of the process regarding different criteria, exemplarily process consistency. Additionally, they support planning of the process. Based on a critical path method, earliest and latest start/end

dates can be calculated. Involved persons can identify their related activities in the process as well as interfaces to other activities or persons based on the assignment of resources.

To support the modeling of processes, ISO/IEC 19510 Part 2.2.3 describes the graphical representation as a BPMN “key element”. In [13] Part 7.3 it is stated, that “Within the basic categories of elements, additional variation and information can be added to support the requirements for complexity without dramatically changing the basic look and feel of the diagram”. Therefore, in context of this extensions and adaptations, different colors and line styles of graphical elements were introduced.

3. BASIC BPMN CONSTRUCTS FOR PROCESS MODELING IN INTEGRATED E/E ARCHITECTURE DEVELOPMENT ENVIRONMENTS

To describe processes in a model-based environment, we focus on the usage of basic BPMN constructs to provide the required modeling artifacts. The standard ISO/IEC 19510:2013 Part 7.3 describes five basic categories of elements: “flow objects”, “data”, “connecting objects”, “swimlanes”, “artifacts”. In context of this approach, the elements “flow objects”, “data” and “connecting objects” are relevant. In order that the benefit of the approach is a first integration into description languages for large-scaled electric electronic architectures and therefore the combination of data and process models in this domain, some adaptations and extensions to the available meta model specification of BPMN [13] were introduced due to domain-specific boundaries in the targeted environment.

3.1 Process Modeling Artifacts

We provide proposed class diagrams for the modeling of process artifacts, which are close to the meta model description of BPMN. These are enriched with additional classes and attributes according to the “BPMN Extensibility” [13] Part 7.7 to fulfill our specific needs. Thus they are compliant with a valid BPMN core.

A super class for all elements which are contained in a process flow is described by the class **FlowElement**, as shown in Figure 2. This concept was extended based on [13] Part 8.4.7. All classes in context of the process flow inherit from **FlowElement** which are namely: *Activity*, *Event*, *DataObject*, *DataStore*, *AbstractPort* and *Connection*. **FlowElement** provides the public attributes *name*, *id* and *description* which are inherited for all sub-classes. The data type for these attributes is the type *string*.

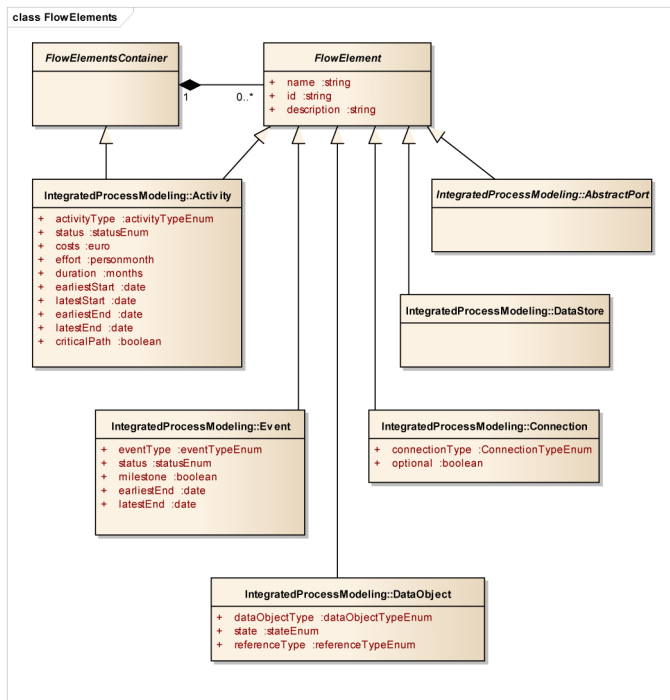


Figure 2. Proposed class diagram 'FlowElements' for the definition of flow elements.

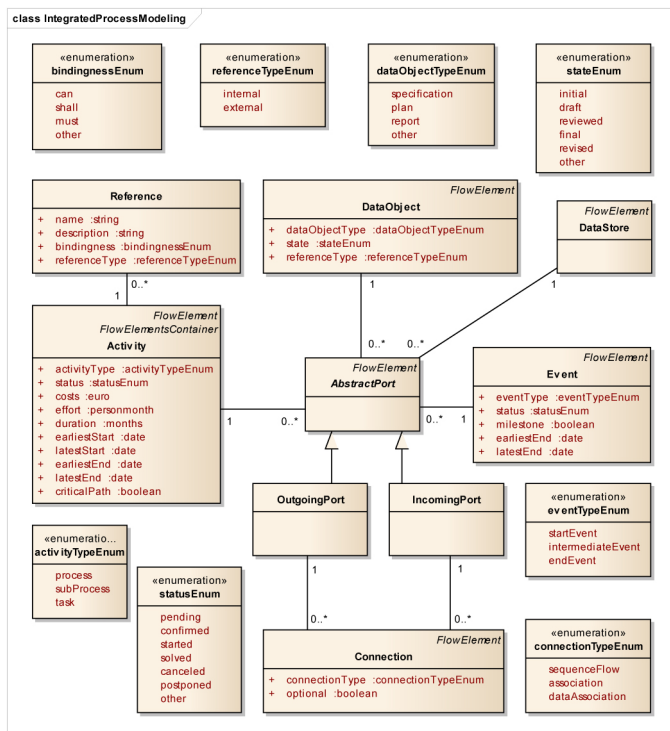


Figure 3. Proposed class diagram 'IntegratedProcessModeling' for modeling process artifacts.

The abstract class **FlowElementContainer** represents a container in which several flow elements can be composed. Therefore, the multiplicity is '0..*'. This class facilitates a hierarchy of processes and sub-processes by the definition of

subsets in context of activities. The **FlowElementContainer** is based on [13] Part 8.4.8 and was extended regarding sub-classes and inheritance.

According to ISO/IEC 19510:2013 Part 7.3, "flow objects" are the "main graphical elements to define the behavior of a process". The following three flow objects are present: "events", "activities" and "gateways". This approach focuses on the usage of activities and events as they represent the basic elements of a process. Simple gateways could be represented using our concept of incoming and outgoing ports. The class diagram for integrated process modeling is shown in Figure 3.

Class Activity: Activities generally describe productive work that is performed in a process [13] Part 10.3. An activity can contain other elements of a process flow and therefore represent a process, sub-process or an atomic work as a task. This distinction is facilitated by the enumeration *activityTypeEnum*. For the description of processes and sub-processes, the activity is derived from the abstract class **FlowElementsContainer**. In accordance to [13] Part 8.4.8, the class describes a containment of specific **FlowElements** such as activities, events and connections via a composition. This supports the description of hierarchy for processes and sub-processes. The attribute *status* indicates the status of an activity, exemplarily if it is started or already solved. Business relevant information such as *costs*, *effort* and *duration* of the activities can be specified in the corresponding attributes. Therefore, specific data types were introduced such as currency *euro*, *person month* or *months* as an extension to the primitive types. The activity is further extended with attributes regarding temporal information such as *earliestStart*, *latestStart*, *earliestEnd* and *latestEnd* which are all of data type *date*. The values of these attributes facilitate analysis and planning of the process after modeling. Activities contained in the critical path of a process can be specified by the boolean attribute *criticalPath*.

The class **Reference** is associated to the class activity and represents a relation to exemplarily standards or other documents with further supporting information. Therefore, this class contains a *bindingness* to express the obligation of the reference. Different obligations exemplarily *must* or *shall* can be expressed using the enumeration *bindingnessEnum*. Due to the fact that this class is not a flow element, the attributes name and description are contained within the class itself. Additionally, the *referenceType* describes the reference source as *internal* or *external* to the process. Each activity can have several references, thus the multiplicity is '0..*'.

Class Event: An event represents "something that happens" during a process and directly affects the flow of a process [13] Part 10.5. The three different types of events are indicated by the attribute *eventType* using the enumeration literals: *startEvent*, *intermediateEvent* and *endEvent*. Similar to the activity, the *status* describes if the event exemplarily is pending. An event can be tagged as a milestone if it is of special importance for the process. The corresponding attribute