Reference Architecture Model Automotive (RAMA)[1]

Version 0.2, 20.04.2017

1 Introduction

RAMA is a graphical representation to model vehicular information and communication technology of (connected) vehicles. It models the integration of vehicles in their environment as well as the whole vehicular life cycle from the beginning of the development till scrapping. RAMA is an adaptation of the Reference Architectural Model Industrie 4.0 (RAMI 4.0) of the VDI, VDE and ZVEI for the automotive domain [1]. RAMA was designed to model use cases, functional behaviour, data, data flows and physical components in a common graphical notation. Though, RAMA is primary constructed to model technical behaviour, automotive business issues can be depicted, as well.

RAMA was designed within the Security Sub-Working Group „Connected and Automatic Driving“ of the Governmental Department of Transport and Infrastructure (BMVI) in Germany, 2016. RAMA is intended to be applied for a security threat analysis concerning automated and connected vehicles.

2 Description

The Reference Architecture Model Automotive includes three dimensions: Automotive layers, automotive hierarchy levels, and the automotive life cycle. The automotive layers are separated in one business layer and four different technical layers:

Layers:

- Business: Organization and business processes
- Function: Elementary function and composition of functions including functional dependencies (e.g., Electronic Stability Control (ESC), Adaptive Cruise Control (ACC))
- Information: Data (internal - and external vehicle data, e.g., from the backend)
- Communication: Communication channels and data flow

1 Defined by the Security Sub-Working Group „Connected and Automatic Driving“ of the Governmental Department of Transport and Infrastructure (BMVI) in Germany, 2016

- Component: Physical components plus firmware

Life cycle:

- Development: Design and production including type-approval
- Operational time: Usage of vehicle
- Scrapping: End of live cycle. Scrapping can have different characteristics: Scrapping of a physical component of a vehicle, scrapping of the whole vehicle or only scrapping of vehicular data due to a selling of a vehicle.

Initially, RAMA is a generic model. So according to [2], a 'Type' can be a 'Fahrzeugtyp', a 'Variante' or a 'Version' of a vehicle of any class 'Klasse': motor vehicle {M, N}, tractor {T, C}, two- or three-wheel vehicles and quadricycles {L} or trailers {O, R, S}. An 'Instance' is one single vehicle of the addressed 'Type'.

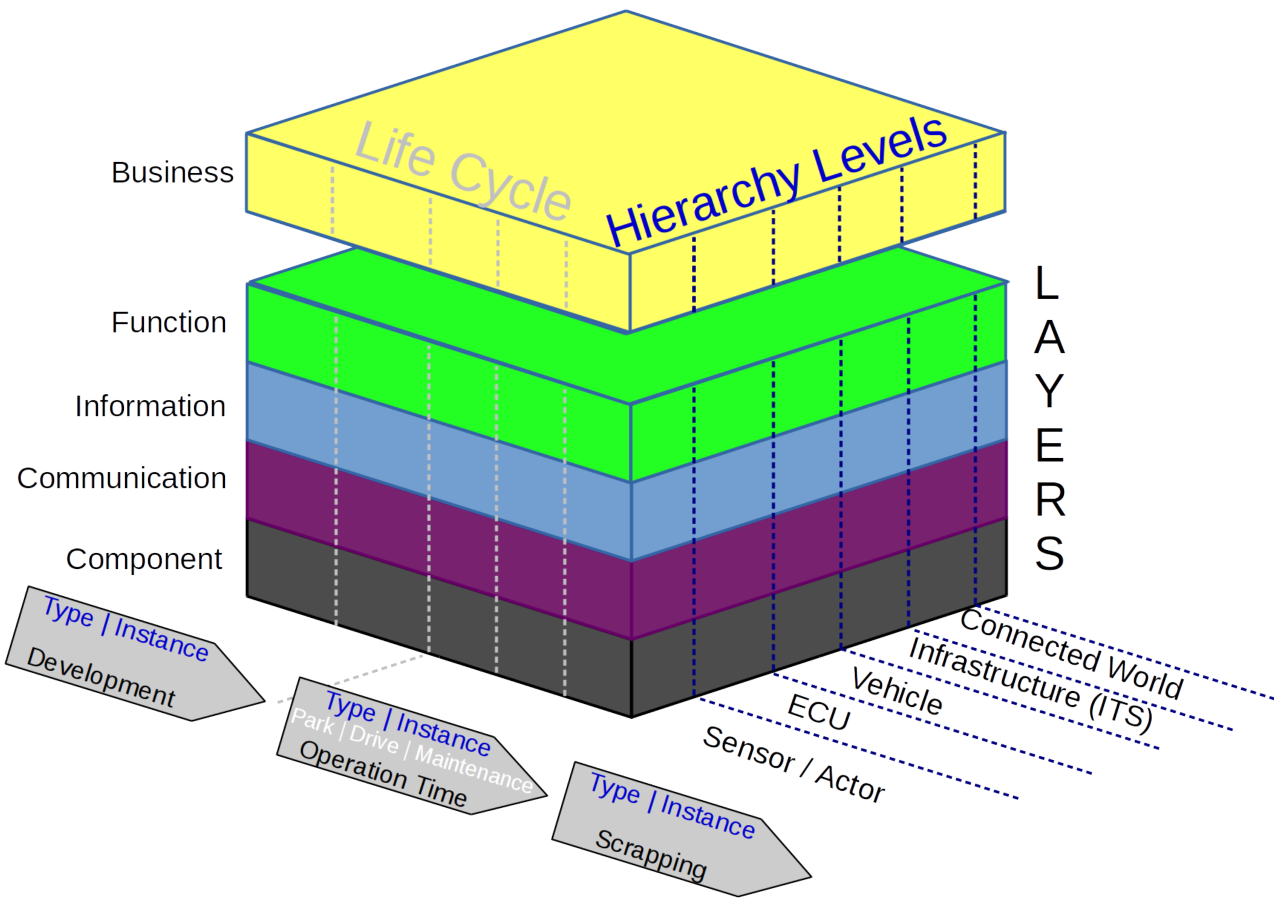


Figure 1, RAMA

Hierarchy Levels:

- Sensor / Actor: This level refers to a individual sensor or actor or a collection of sensors and/or actors of a vehicle.
- ECU: Individual ECU or collection of ECUs.
- Vehicle: This refers to the complete IT of a vehicle including sensors, actors, ECUs, the internal networks and internal/external communication units.
- Infrastructure: A vehicle may interact locally with external components of the road side or maintenance infrastructure (or other vehicles), so this level covers all types of vehicle-to-X communication, e.g., at a garage.
- Connected world: This level covers all types of communication via internet including connections to the OEM backend or other services (maps, traffic info, etc.)

3 Modelling and Views

RAMA can be partitioned into views. In Figure 2, the general approach to generate views based on separation or sectioning of the three dimensions life cycle, hierarchy level and layers is given. Figure 3 shows a “vehicle development view“.

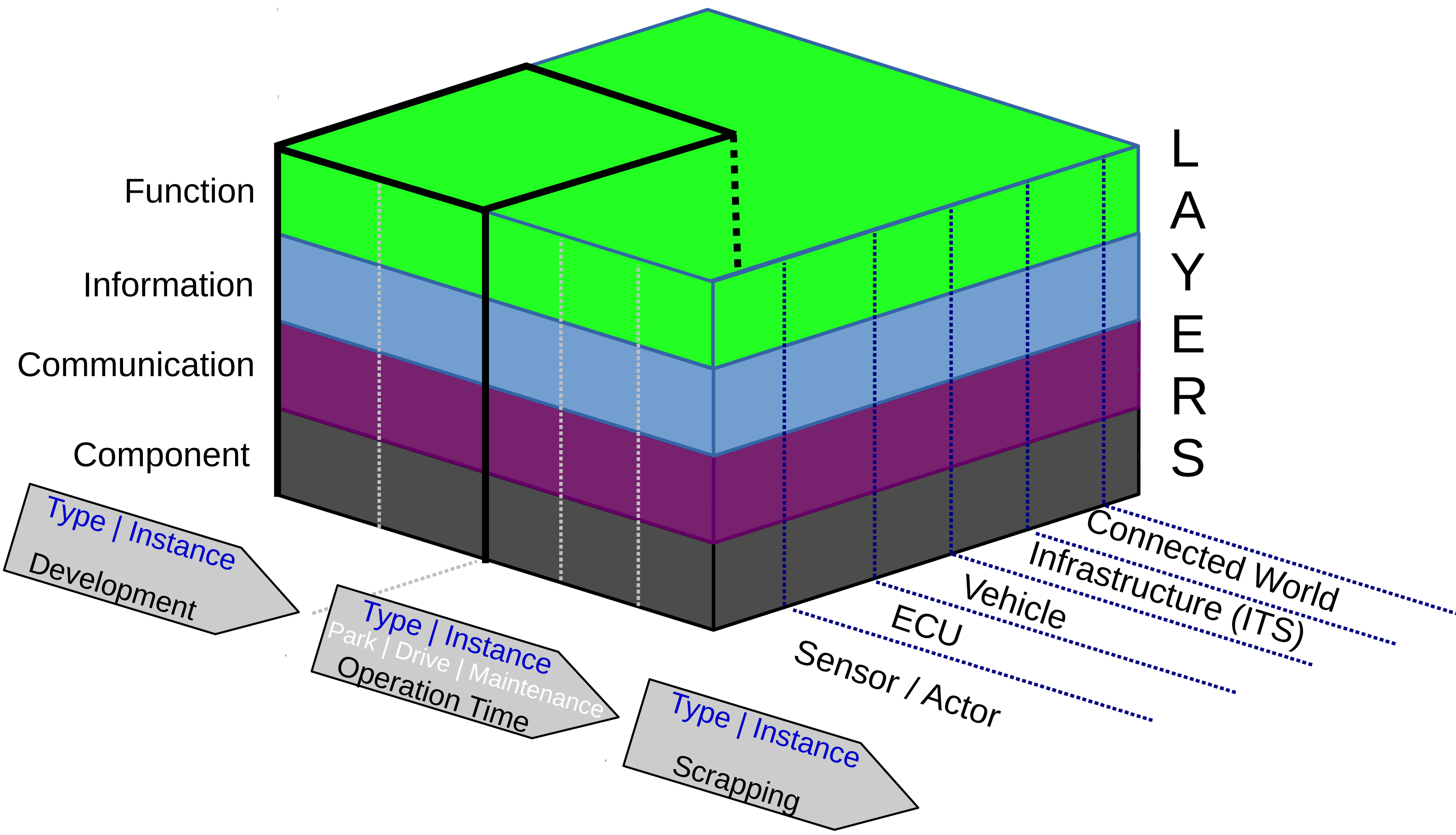


Figure 2, RAMA separations to generate views

3.1 Vehicle Development View

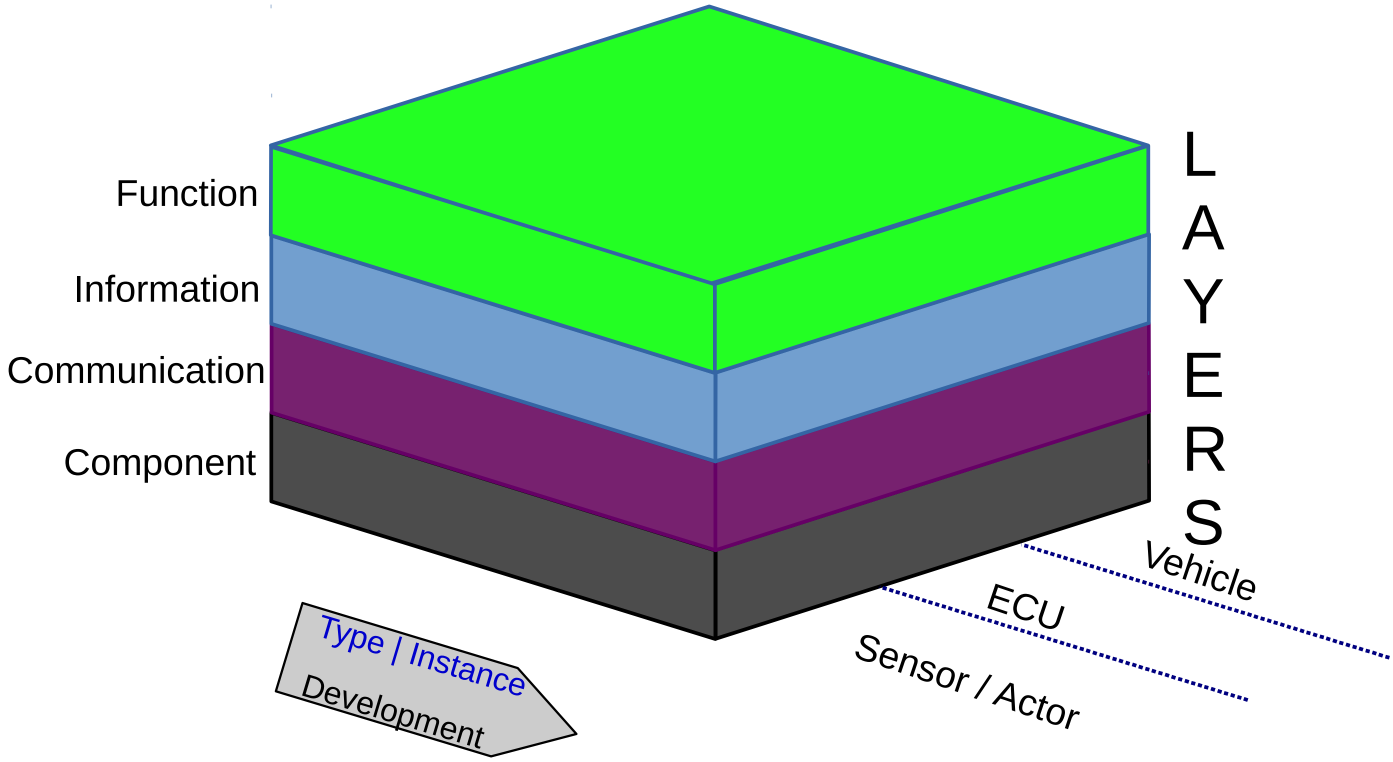


Figure 3, vehicle development view

3.2 Function Domains

Next, function domains are introduced to model the separation of components, functions as well as the relevant data, and communications.

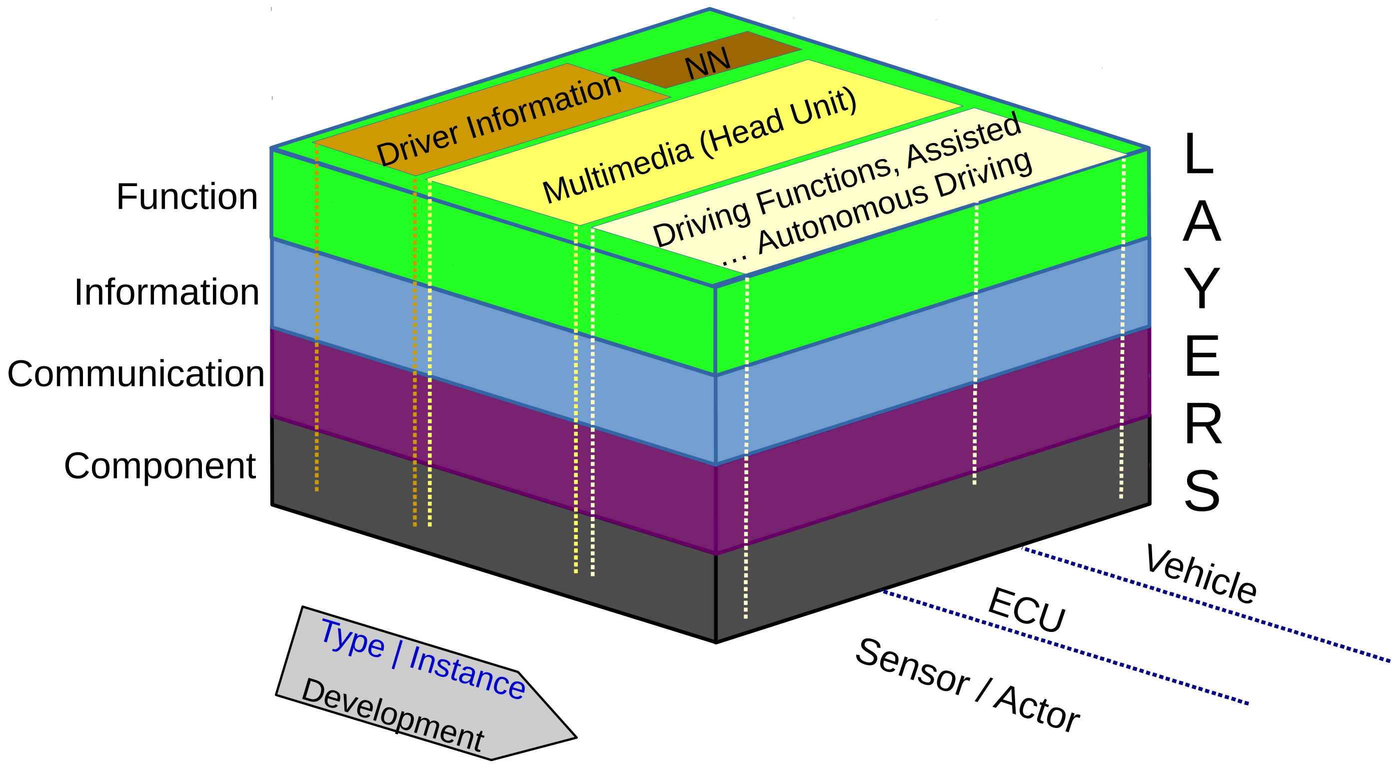


Figure 4, vehicle function domains

In Figure 4, four different example function domains are depicted. These are :

1. Necessary components for driving and its functions including assisted and automatic driving,
2. Components and functions to display driving information to the driver,
3. Multimedia components and its functions and services,
4. NN (not defined).

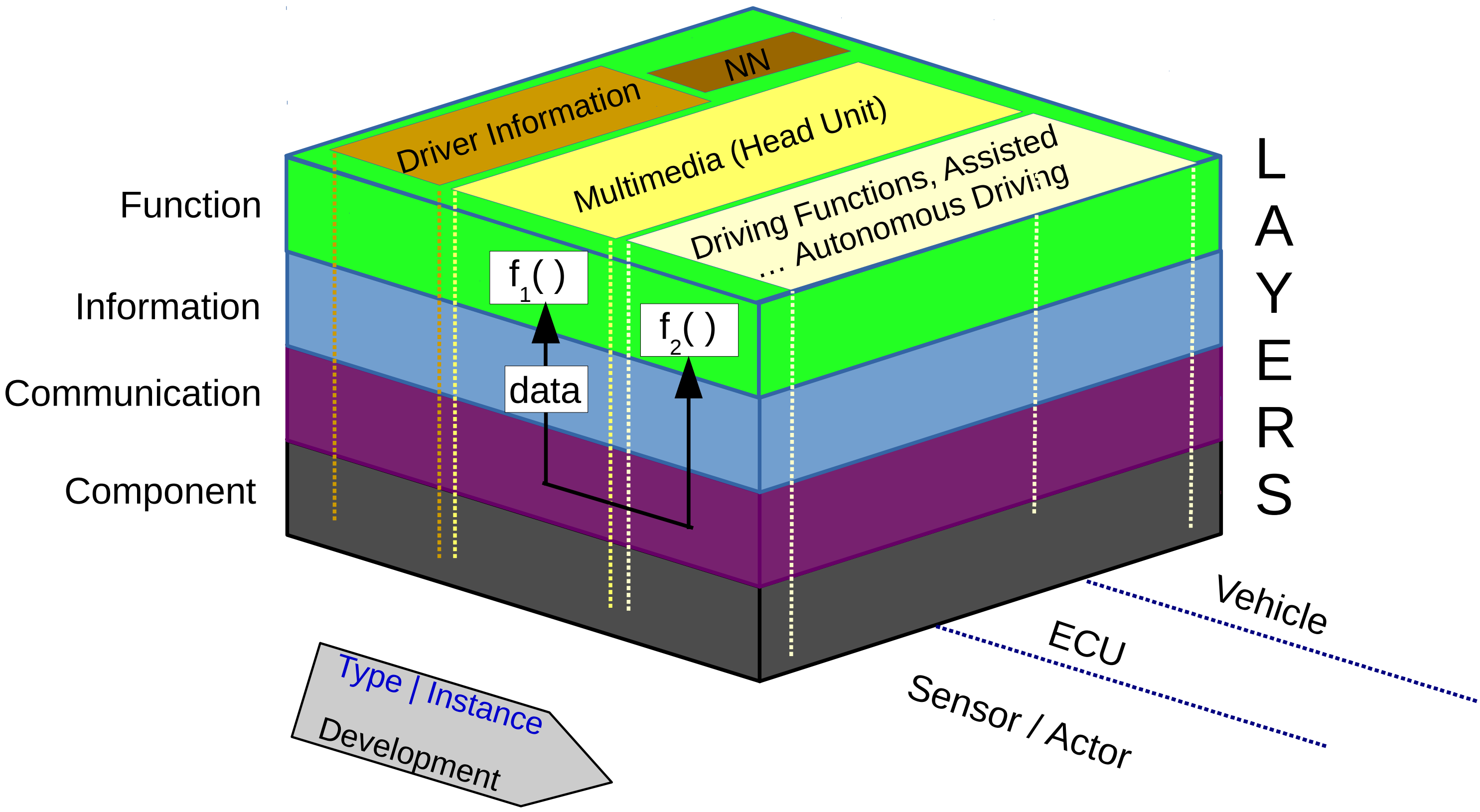


Figure 5, information flow between domains

Function domains are not strictly separated on the lower layers. Interactions and communication flows between these domains are possible. E.g., in Figure 5 the exchange of data between the functions *$f_1()$* and *$f_2()$* is shown without any specification of the concrete data type or interfaces of the functions *$f_1()$* respective *$f_2()$* (for example the assisted/autonomous driving functionality may indicate the current status or planned driving routes on the head unit display).

4 Application Example

4.1 Data Logger

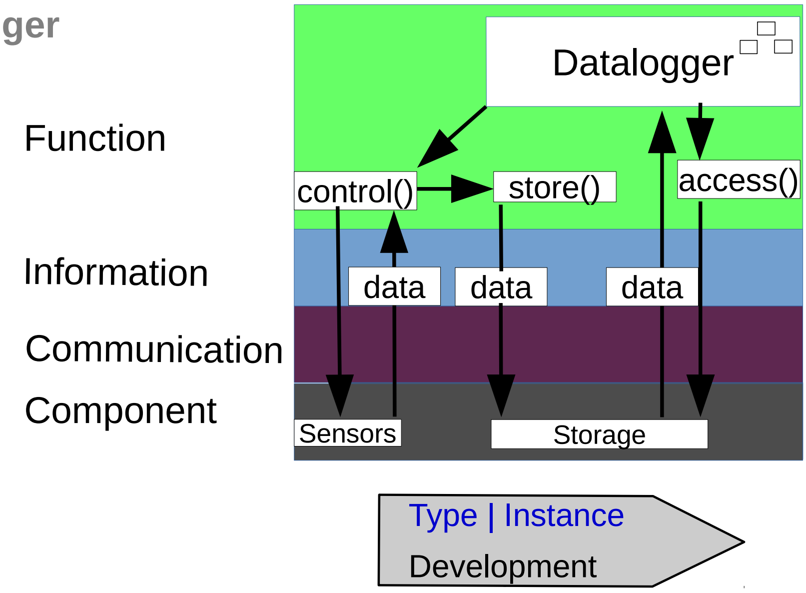
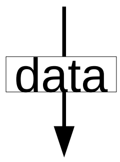


Figure 6, brief functional behaviour of a data logger

Figure 6 models a brief graphical representation of a top-level function 'Datalogger'. The function 'Datalogger' is an event data recorder for automated driving. Aim of these function is to monitor and store vehicular sensor information as well as accessing already stored data. Here, neither the events which should be stored nor access rights or any other security functionality is specified.


Arrows indicate control flow and/or information flow.



An arrow combined with a “data box“ explicitly address information flow

The top-level function 'Datalogger' is comparable to a *CompositionType*, the functions: 'control()', 'store()' and 'access()' are comparable to an *AtomicSoftware-Component Type* of AUTOSAR [3].

5 Threat Analysis

To be done.

6 Future Issues

As shown in Figure 6, graphical notations of [3] can already be used in RAMA. But an unified semantical integration of existing graphical notation in RAMA is an open issue.

7 References

[1] ZVEI, “The Reference Architectural Model Industrie 4.0 (RAMI 4.0)“, http://www.zvei.org/Downloads/Automation/ZVEI-Industrie-40-RAMI-40-English.pdf

[2] Kraftfahrt-Bundesamt (KBA), Document: 2007L0046 – DE, Seiten 114 – 121, 04.02.2015,

[3] AUTOSAR, Specification of Graphical Notation, V 1.0.1, 2006