Annex B
(informative)

Example architectures for Coexistence of elements and Decomposition of requirements

B.1 Example architecture

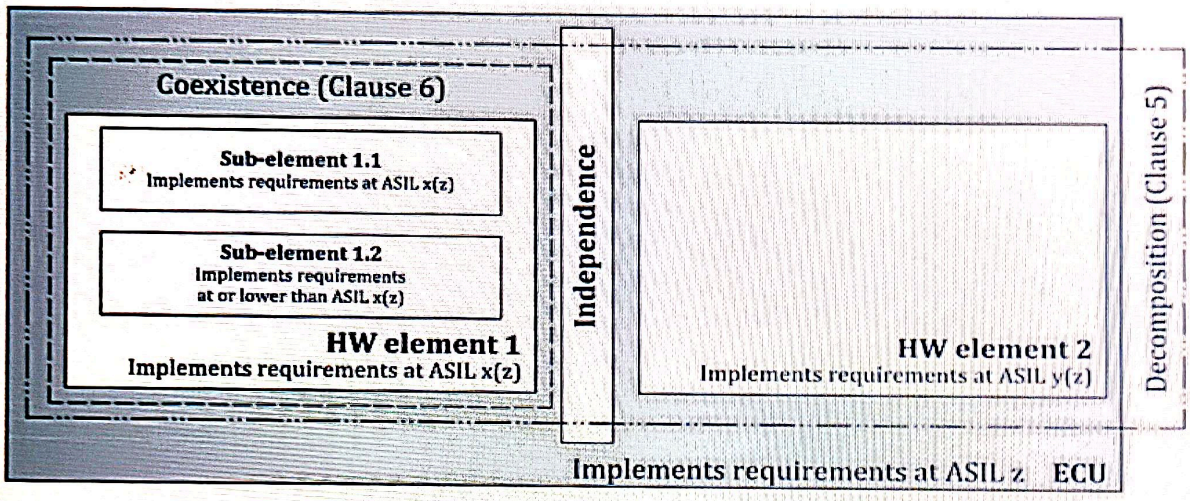

Figure B.1 — Coexistence and decomposition in an example architecture

NOTE Coexistence and decomposition are part of the system architectural design constraints addressed in ISO 26262-4:2018, Clause 6; hardware design in ISO 26262-5:2018, Clause 7; and software in ISO 26262-6:2018, Clause 7.

B.2 Coexistence (Clause 6):

— If an ASIL x requirement is allocated to Element 1, then Sub-element 1.1 and Sub-element 1.2 inherit ASIL x.

— Sub-element 1.2 is only developed at a lower ASIL if the following conditions are met:

  — At least one sub-element of Element 1 is able to fulfil Element 1 requirement at ASIL x (e.g. Sub-element 1.1);

  — Sub-element 1.2 cannot violate Element 1 safety requirement; and

  — Criteria for coexistence are met (see Clause 6): no cascading failures from Sub-element 1.2 to Sub-element 1.1 (Freedom From Interference).

B.3 Decomposition (Clause 5)

If an ASIL z requirement is allocated to the ECU in Figure B.1, it can be decomposed between independent AND redundant hardware elements.

This is equivalent to meeting all of the following conditions:

— A decomposition schema described in Clause 5 is used, i.e. ASIL z → ASIL x(z) + ASIL y(z);

— HW_Element_1 fulfils by itself the ECU safety requirement at ASIL x(z);

— HW_Element_2 fulfils by itself the ECU safety requirement at ASIL y(z); and

— HW_Element_1 and HW_Element_2 are independent, i.e. no cascading failures from HW_Element_1 to HW_Element_2 and no cascading failures from HW_Element_2 to HW_Element_1, no common cause failure, demonstrated at ASIL z.

Annex C
(informative)

Framework for Identifying Dependent Failures

Independence between two or more elements is determined by showing absence of dependent failures, i.e. absence of cascading failures and common cause failures. Independence can be required between elements according to the safety concept, e.g. to support ASIL decomposition.

In order to identify cascading and common cause failures, the following classes of coupling factors can be used to improve the completeness of the analysis.

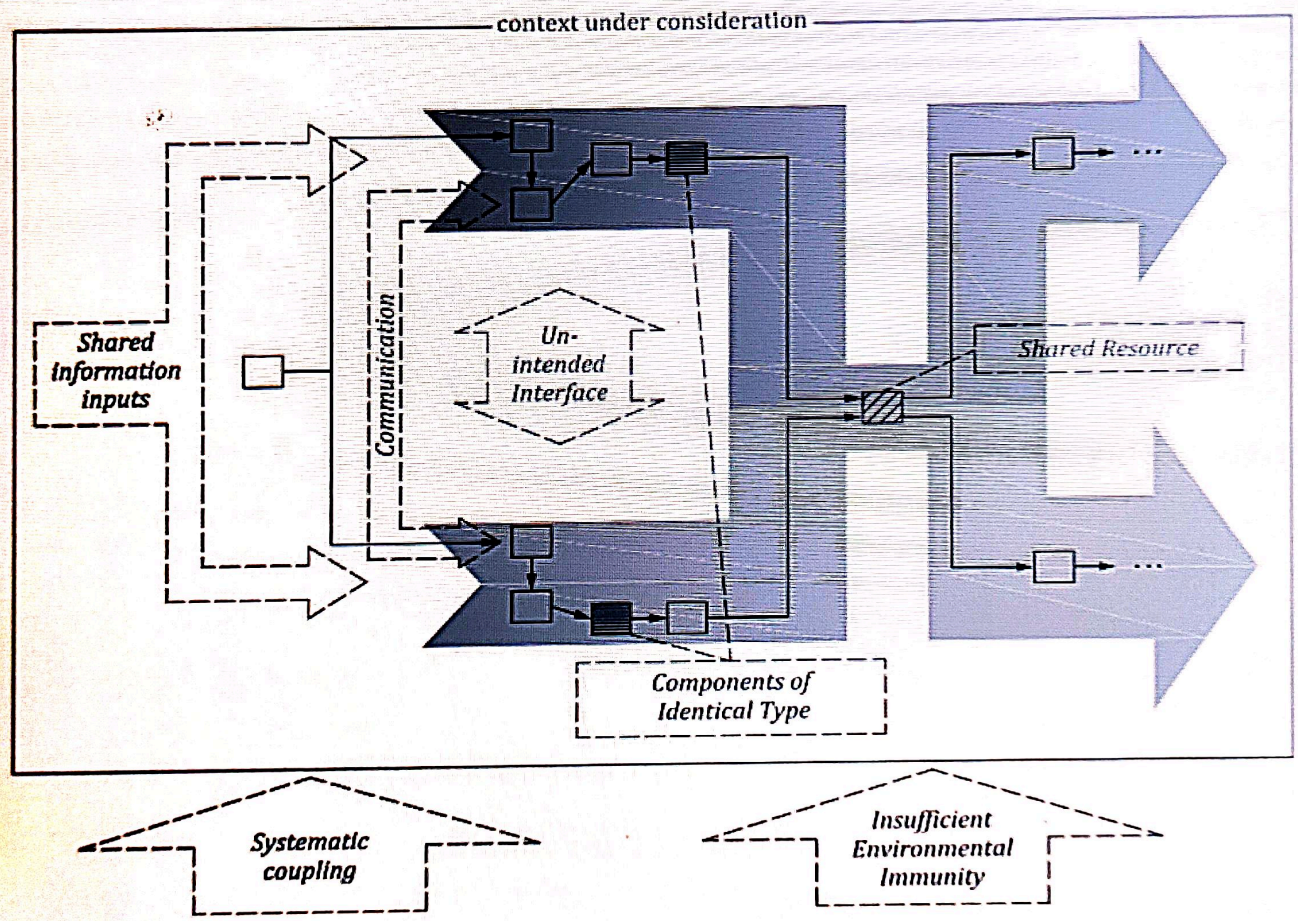

Figure C.1 — Coupling factor classes between elements

NOTE The grey arrows indicate the functional chain of events linking the elements thereby realizing the functionality affected by coupling factors. The dotted arrows indicate the classes of coupling factors potentially affecting the system and its elements.

These classes of coupling factors can be applied as checklists to any level of abstraction, including the system, software, hardware, and semiconductor levels, as illustrated in Table C.1. This table presents examples of coupling factors, which are mapped to the topics in 7.4.4. Some examples can belong to several coupling factor classes, e.g. software calibration parameters might be considered as Shared Resource or Shared Information Input.

Table C.1 — Examples at the system, software, hardware, and semiconductor levels

Coupling factor class	Mapping to the topics in ISO 26262-9:2018, 7.4.4	Examples at the system level	Examples at the hardware level	Examples at the software level	Examples at the semiconductor level
Shared Resource The same software, hardware, or system element instance is used by two elements, which are therefore affected by the failure or unavailability of that shared resource.	a) random hardware failures g) failures of common external resources or information	— Power supply (see also Insufficient Environmental Immunity) — Wiring harness — Data and communication busses — Powerstage	— Clock — Same H-Bridge used by two shut-down paths — Sockets, plug connectors	— SW component used by 2 other SW components, e.g. maths or other libraries — I/O routines, drivers — Hardware resource used by more than one software element	"Failure of shared resources" and "single physical root cause" in ISO 26262-11
Shared Information Input Connection to the same information source by means of which the two functions consume the same information, even in absence of shared resources, i.e. from a functional perspective.	a) random hardware failures	— External messages (e.g. CAN, Flexray, or AUTOSAR RTE messages) — External physical signals (e.g. magnetic fields, remote/radio signals) — Readings detected by capacitive/radar/optical sensors	— Connection to sources of raw physical digital or analogous signals	— Constants, or variables, being global to the two software functions — Data/function parameter arguments/messages delivered by software function to more than one other function	"Failure of Shared resources" in ISO 26262-11
Insufficient Environmental Immunity Same or similar physical characteristics of elements, which can be affected by the same external environmental disturbance	f) environmental factors h) stress due to specific situation	— Mechanical coupling — Flammable material	— Grade of sensitivity to electrical effects (creating the potential of suffering from e.g. EMI, ESD) — Proximity of HW elements (creating the potential of suffering from e.g. dust, particles) — Same housing (creating the potential of suffering from e.g. water entry, humidity)	Not directly applicable to SW alone. Environmental influences that affect behaviour of software can be considered at the system and hardware levels	"Environmental faults" in ISO 26262-11

Table C.1 *(continued)*

Coupling factor class	Mapping to the topics in ISO 26262-9:2018, 7.4.4	Examples at the system level	Examples at the hardware level	Examples at the software level	Examples at the semiconductor level
Systematic Coupling Failure of elements due to a common systematic human error or tool error.	b) development faults c) manufacturing faults d) installation faults e) service faults h) stress due to specific situation	— Identical production processes used for multiple elements — Identical repair processes used for multiple elements.		— Same software tools e.g. IDE, compiler, linker, software configurator — Same programming and/or modelling language — Same compiler/linker	"Development faults", "Manufacturing faults", "Installation faults", and "Repair faults" in ISO 26262-11
Components of Identical Type Multiple instances of identical or very similar components can jointly fail due to a common cause failure.	a) random hardware failures b) development faults	— Same type of actuator/power stage e.g. motor — Same type of sensors	— Same type of HW parts and components — Same power supply ICs for different microcontrollers — Same microcontroller — Same ASICs	— The same source code expanded twice, e.g. by usage of C macros NOTE: the same library instance or the same standard SW module instance called from different locations is rather considered as a Shared Resource.	"Development faults" in ISO 26262-11
Communication An element receives information from another element by means of a communication channel	a) random hardware failures b) development faults d) installation faults e) repair faults i) ageing and wear	— CAN connection between two ECUs of the same system — Communication between two microcontrollers within the same ECU	— Electrical connection between two HW elements	— Data flow via global variables — Messaging — Function calls with arguments passed	"Failure of shared resources" and "single physical root cause" with semiconductors in ISO 26262-11

Table C.1 *(continued)*

Coupling factor class	Mapping to the topics in ISO 26262-9:2018, 7.4.4	Examples at the system level	Examples at the hardware level	Examples at the software level	Examples at the semiconductor level
Unintended Interface Two elements affecting each other directly via an unanticipated interface	a) random hardware failures b) development faults d) installation faults h) stress due to specific situation	— One functionality overruling the other because of missing synchronization	— Proximity of HW elements, creating the potential of crosstalk between signal lines, heat impact, interference etc.	— Same memory space which means a potential of wrong memory allocation or memory leaks	"Single physical root cause" originating from a semiconductor (see ISO 26262-11)