THE USES AND ABUSES OF ASIL DECOMPOSITION IN ISO 26262

D.D. Ward*, S.E. Crozier†

**MIRA Limited, UK, david.ward@mira.co.uk, †MIRA Limited. UK, steve.crozier@mira.co.uk*

Keywords: ISO 26262, requirements, architecture, ASIL.

Abstract

This paper examines the ISO 26262 approach to ASIL decomposition, more appropriately called "requirements decomposition", and how it may be applied correctly during the requirements analysis and architectural design of a safety-related automotive control system.

1 Introduction

ISO 26262 [1] was published in November 2011 as a functional safety standard for electrical and electronic systems in road vehicles. It claims to be the automotive version of IEC 61508 [2]; although in reality, while it applies many of the same principles there are also significant differences, and it would be more accurate to describe it as an interpretation rather than a sector implementation of IEC 61508.

Nevertheless the standard has addressed many of the issues that would be encountered in endeavouring to apply IEC 61508 directly to automotive products [3], and has a number of strong features. Amongst these are:

- A well-defined hierarchical approach to safety requirements specification and derivation;
- The need to carry out detailed safety analyses to understand and defend against failure modes of the system under development that could lead to hazards;
- The need to understand architectural constraints such as ensuring adequate diagnostic coverage by and of safety mechanisms that defend against these failure modes, and avoiding failure mode propagation and common-cause failures between the constituent elements of the system.

One of the techniques that can be applied during architectural system design is called "requirements decomposition".

2 What is requirements decomposition?

When applying ISO 26262, a hazard analysis and risk assessment is carried out at the level of the "item" (essentially the system under development) and a set of "safety goals" are specified — these are very high level safety objectives for the item, typically expressed in terms of preventing or mitigating a hazard at the vehicle level; and these safety goals are assigned the ASIL value resulting from the risk classification of the associated hazard.

The "Automotive Safety Integrity Level (ASIL)" value represents the degree of rigour that should be applied in development, implementation, and verification of a requirement in order to avoid unreasonable residual risk in the final product.

As application of the process required by the standard progresses, safety goals are successively refined through a hierarchy of safety requirements including functional safety requirements, technical safety requirements and eventually hardware and software safety requirements. This is intended to reflect an iterative approach, based on a classical "V" model, to refine high-level requirements into implementation-specific low-level requirements. During this process, safety requirements are allocated to elements of the architectural design. Each safety requirement inherits the ASIL value of its parent in the hierarchy; and ultimately the ASIL value of the safety goal(s) from which it is derived.

However, if in the design of the architecture sufficiently independent and redundant elements exist, then it is possible to allocate a specific safety requirement to two (or more) of these elements. The redundant requirements so allocated may then inherit a lower ASIL value than the parent. For example an ASIL D safety requirement may be allocated to two independent architectural elements; the requirements allocated are then called "decomposed" requirements and can inherit a lower ASIL value than the parent, such as ASIL B.

However the important point to understand is that the requirements so decomposed are effectively still **the same requirement** as the parent requirement as illustrated in Figure 1 below:

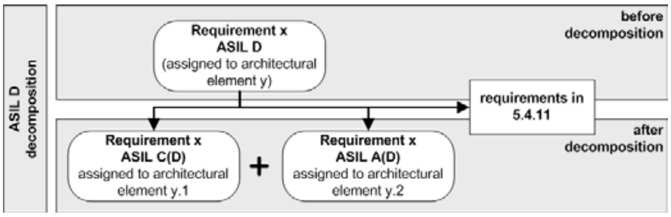

Figure 1: Application of requirements decomposition

Due to the fact that the ASIL values of derived requirements may be reduced in this way, the technique is often (incorrectly) called "ASIL decomposition". Whilst

ISO 26262 does from time to time refer to "the application of ASIL decomposition" along with "ASIL tailoring" the more appropriate term is "**requirements decomposition** with respect to ASIL tailoring" (i.e. the title of Part 9 Clause 5 of the standard). Unfortunately the "ASIL decomposition" term has entered common use (hence the title of this paper), but it is more accurate to describe it as "requirements decomposition" since it is the requirements that are manipulated. As a result of the requirements manipulation the ASIL value may then be reduced. The ASIL value is not manipulated directly.

3 Misinterpretations of "ASIL" decomposition

This incorrect terminology is unfortunately associated with a great deal of misunderstanding about the purpose and application of the technique. In particular it is often assumed that:

- Requirements decomposition (especially when misinterpreted as ASIL decomposition) is a "must do" requirement when applying the standard. In fact this is not the case; there is no obligation to apply requirements decomposition when claiming compliance with ISO 26262; but **if** it is applied **then** certain additional requirements have to be complied with. These requirements are given in Part 9 Clause 5 of the standard and are further explored in Section 4 below.
- ASIL decomposition is frequently misinterpreted as an objective; in other words, a frequently encountered (and incorrect) question is "There is an ASIL D safety goal; now how can it be decomposed into ASIL B elements?" It is not valid to create an element out of sub-elements with lower ASIL values through such a "building block" approach without considering the independence of the redundant elements and their associated safety requirements (i.e. without considering the suitability of the architecture to support this).

Below are some examples illustrating common misuse of requirements decomposition.

Example 1: Electronic Steering Lock

In this example the item is an electronic steering lock; the purpose being an anti-theft device to drive a locking bolt into the steering column when the vehicle is locked, thus preventing unauthorized operation.

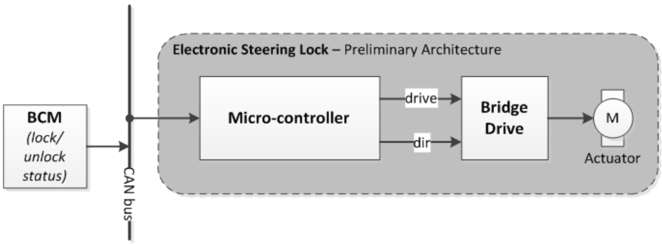

Figure 2: Electronic Steering Lock – preliminary architecture

In this example, a message is received on the CAN bus, processed and validated by the microcontroller to drive the bridge and thus the motor to operate the bolt in an appropriate direction, at an appropriate time, to lock or unlock the steering column. Amongst others, a safety goal is given:

SG01: When the vehicle is being driven, the steering lock shall not engage unintentionally [ASIL D].

Through development of the "Functional Safety Concept" and then the "Technical Safety Concept" the developers have established that given other project constraints use of the microcontroller and software to control the lock may not capable of fulfilling the technical safety requirement below to the given ASIL:

REQ 22: The Microcontroller shall activate the drive signal to the bridge drive when "lock" conditions are received over the CAN bus. [ASIL D]

As a result, the architecture has been updated to add a second microcontroller and requirements decomposition has been attempted.

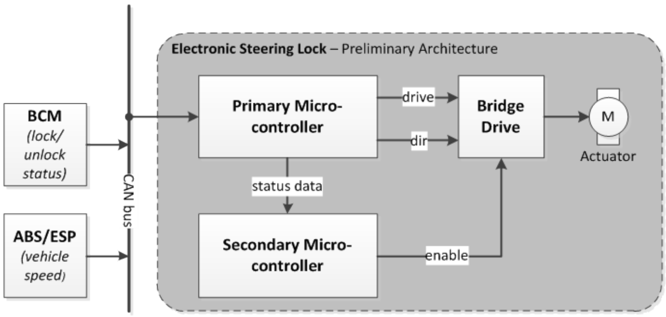

Figure 3: Electronic Steering Lock – flawed decomposition

REQ 22.1: The primary microcontroller shall activate the drive signal to the bridge drive when a "lock" command is received over the CAN bus. [ASIL B(D)]

REQ 22.2: The secondary microcontroller shall activate the enable signal to the bridge drive when the vehicle speed received over the CAN bus indicates that lock conditions are plausible (i.e. the vehicle is stationary). [ASIL B(D)]

With the requirements decomposition in place, the expectation was that the primary microcontroller element can now be developed to ASIL B(D) since B(D) would be the highest level of integrity for requirements that need to be realized by this element.

What has been neglected here is that the primary microcontroller is acting as a gateway to receive the vehicle speed CAN message and pass this onto the secondary microcontroller. This means that there is a common-cause failure (the CAN reception; element's sub-parts and software) which could cause both REQ 22.1 and REQ 22.2 to fail simultaneously. *Furthermore, failures associated with the CAN bus itself have not been considered, or the ability for one of the other CAN nodes to spoof messages, although arguably while a security requirement this is outside the scope of ISO 26262.*

Figure 4 illustrates a plausible solution to this problem, where the secondary micro is also equipped with CAN hardware and therefore is able to fulfil REQ 22.2 independently from the primary microcontroller. Furthermore, it is noted that a separate CAN bus is also utilized to avoid common-cause failures external to the item. In addition, utilizing diverse sources of inputs means that the integrity of those inputs need not be so great, so such a solution avoids the need to place such high integrity requirements upon external systems.

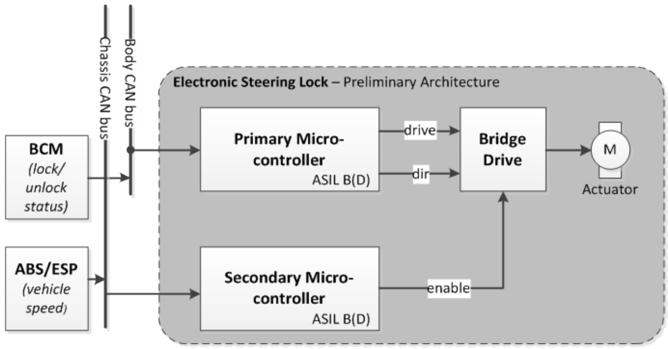

Figure 4: Electronic Steering Lock – plausible decomposition

Finally, as a cautionary note; one often overlooked aspect is that of support circuitry for the microcontrollers. In such architecture, care should be taken to ensure that there is an absence of common cause faults, and that no propagation of faults from one microcontroller to the other can occur via a common power supply or common clock circuit.

Example 2: Steer-by-wire

In this example (which is intended to illustrate principles and not represent a real design) a 4-wheel steer-by-wire system is under development. Here, the hand-wheel input is sensed and processed into commands for control of the front and rear axle actuators. A third actuator provides haptic feedback to the driver at the hand-wheel.

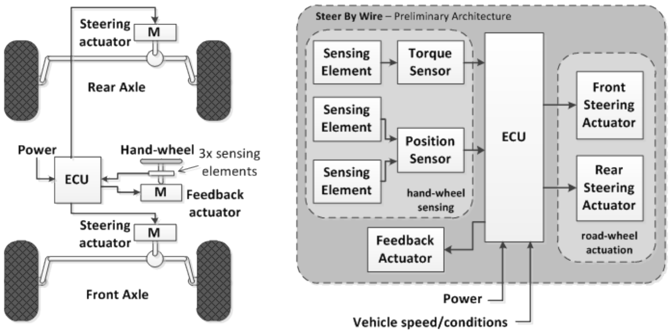

Figure 5: Steer-by-wire system

A number of safety goals have been established for this system which when flowed-down result in the following requirements being placed upon the hand-wheel sensing and road-wheel actuation (amongst other elements of the item):

REQ 32: Failure of the road-wheel actuation shall not lead to an absence of directional control of the vehicle. [ASILD]

REQ 49: Failure of the hand-wheel sensing shall not lead to an incorrect indication of the driver's intended direction to the ECU [ASIL D]

Immediately on seeing these ASIL D requirements being placed upon sensing and actuation elements, the first reaction was to decompose. The following requirements were therefore proposed for the actuation output:

REQ 32.1: The front road-wheel actuation shall provide directional control of the vehicle according to ECU commands. [ASIL C(D)]

REQ 32.1: The rear road-wheel actuation shall provide directional control of the vehicle according to ECU commands. [ASIL A(D)]

At first sight, since there are two actuators one might consider an inherent redundancy in the system, and that the original requirement (REQ32) can be fulfilled by either of the two actuators, thus we have a means of decomposition.

However, on closer examination this is not the case. If one axle-actuator fails at a high steering angle, and remains fixed in that position then with the other axle-actuator full steering capability will not be possible. Furthermore, maintaining a straight trajectory of the vehicle would only be possible if the vehicle were slewing diagonally across traffic lanes. Considering such decomposition would at least require a re-work of the hazard analysis, or as an alternative place requirements for additional mechanisms to return the failed axle-actuator to a straight ahead position.

Furthermore an attempt was made to decompose the hand-wheel sensing requirement:

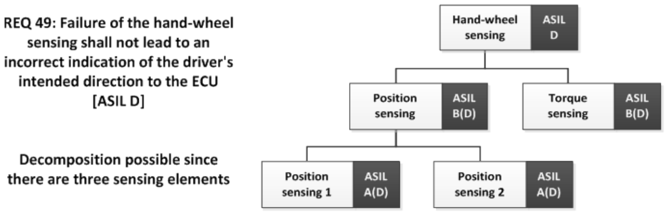
REQ 49: Failure of the hand-wheel sensing shall not lead to an incorrect indication of the driver's intended direction to the ECU [ASIL D]
Decomposition possible since there are three sensing elements
Hand-wheel sensing
ASIL D
Position sensing
ASIL B(D)
Torque sensing
ASIL B(D)
Position sensing 1
ASIL A(D)
Position sensing 2
ASIL A(D)

Figure 6: Hand-wheel sensing "decomposition"

Since there are three sensing elements, then it was considered possible to decompose the sensor into three sensing elements, each carrying lower ASIL requirements.

Whilst this sounds very attractive in being able to develop the system with lower integrity (lower cost) elements, the developer has failed to consider exactly how each of these elements will **independently** fulfil REQ 49.

Taking an example, if REQ 49 is decomposed into:

REQ 49.1 Failure of position sensing shall not lead to incorrect indication of the driver's intended direction to the ECU [ASIL B(D)]; and

REQ 49.2 Failure of torque sensing shall not lead to incorrect indication of the driver's intended direction to the ECU [ASIL B(D)]

it is then established from fault tree analysis that a failure mode of the torque sensor is an offset on the signal output. It is not possible for the ECU to distinguish between the two plausible signals from torque and angle based sensing, one which has an offset and one which does not.

The key problem here is that an attempt has been made to decompose the sensor (the element), whereas the decomposition must be applied to the requirement. A possible solution might be to implement a two out of three voting scheme here, but it is not practical with three sensing element to reduce the requirements as far as A(D) + A(D) + B(D).

4 The benefits of requirements decomposition

It has been established from the previous examples that there are times when some means of requirements decomposition is desirable. This is frequently when constraints on the design mean that specific elements **must** be used; which turn out to be incapable of fulfilling requirements placed upon them at the given ASILs.

Essentially there are two reasons why this might be the case:

- The item has not been developed employing appropriate methods and measures commensurate with a level of rigour in avoidance of systematic faults appropriate to the ASIL; or
- The item is not capable in itself of meeting the relevant targets for performance in regard to handling random hardware failures.

It should be noted that in the first case, homogenous redundancy (duplicating the element) cannot be used without other measures, since common tools, methods and software may lead to common-cause failures, thus the criteria for independence would not be complied with. However, if independent means of fulfilling the requirement are used, and providing the independence criteria can be met then arguably the largest benefit of applying requirements decomposition is in meeting the systematic aspects of safety integrity.

ISO 26262 Part 5 introduces three targets regarding performance in relation to handling random hardware failures. The design must be analysed against these, on a per-safety-goal basis. The benefits may not be initially clear in how applying requirements decomposition can help meet the targets for performance in regard to handling random hardware failures (the “Single Point Fault Metric [SPFM]”, “Latent Fault Metric [LFM]” and Probabilistic Metric for Hardware Failure [PMHF]”). Requirements decomposition is taking advantage of redundancy in the architecture to help meet the PMHF and SPFM by eliminating single point faults, through multiple implementations of a safety requirement. This is illustrated below with a segment of fault tree analysis for the Electronic Steering Lock final implementation example.

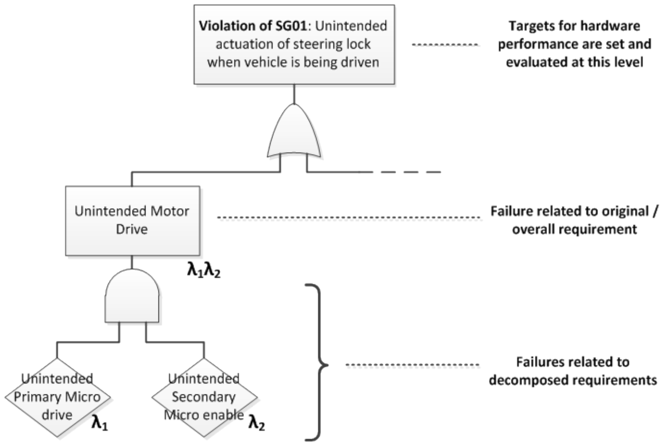
Violation of SG01: Unintended actuation of steering lock when vehicle is being driven
Targets for hardware performance are set and evaluated at this level
Unintended Motor Drive
Failure related to original / overall requirement
$\lambda_1\lambda_2$
Unintended Primary Micro drive
λ_1
Unintended Secondary Micro enable
λ_2
Failures related to decomposed requirements

Figure 6: Effect of requirements decomposition on failure rate

It should be noted (ISO 26262 Part 9 Clause 5.4.5) that the process for evaluation of hardware performance remains unchanged (i.e. it should be conducted per safety goal and at the highest level; not from the decomposed requirement downwards).

There are other motivations behind requirements decomposition, a common motivation being the need to support legacy elements. Here, in a similar structure to Figure 4, a legacy component that may have only be developed to QM (that is developed to established Quality Management processes and principles), may be used to fulfil an ASIL D requirement if this is monitored by a safety mechanism developed to ASIL D(D). This approach can sometimes be more cost-effective in a development programme if avoidance of safety goal violation can be achieved in a straightforward manner by the safety mechanism, as opposed to re-designing the main function which could be implementing substantial additional (and perhaps not safety-related) functionality. Of course, in taking such an approach attention needs to be paid to ensuring freedom from interference, particularly where software elements are concerned.

5 The requirements for applying decomposition

Provided that certain requirements are adhered to then the following ASIL decompositions are permitted:

- ASIL D → ASIL C(D) + ASIL A(D)
- ASIL D → ASIL B(D) + ASIL B(D)
- ASIL C → ASIL B(C) + ASIL A(C)
- ASIL B → ASIL A(B) + ASIL A(B)
- ASIL *x* → ASIL *x*(*x*) + QM(*x*)

The notation B(D) for example shows that the requirement has been decomposed, and the ASIL value of the parent safety goal is shown in brackets.

It should be noted that alternative schemes resulting in a higher decomposition can also be applied e.g.

- ASIL D → ASIL C(D) + ASIL C(D)

As mentioned, in order to apply decomposition a number of requirements must be met:

- Part 9, Clause 5.4.7: If decomposition results in a function plus a safety mechanism then the safety mechanism must carry the higher ASIL; for example as per the example at the end of Section 4 QM(D) + D(D).
- Part 9, Clause 5.4.11a: “Confirmation measures” (independent review) according to Part 2 Clause 6.4.7 must be conducted in accordance with the original ASIL of the safety goal.
- Part 9, Clause 5.4.11b: Evidence for sufficient independence of the elements involved (after decomposition) must be made available (meaning an analysis of dependent failures is required as per Part 9 Clause 7).

In addition, Part 9 Clause 5.4.12 calls specifically when decomposing an ASIL D requirement into ASIL B(D) requirements that additional constraints are imposed (primarily to avoid introduction of systematic failures):

- The decomposed requirements (which are effectively at ASIL B) must be specified using the ASIL C level of rigour which is normally associated with the use of more formalized notations.
- If the same software tools are used for development of the decomposed elements then these must be considered as tools for developing ASIL D requirements and the commensurate confidence in their use established (see Part 8 Clause 11).

All integration activities must be conducted in accordance with the requirements of the standard to the original ASIL, that is, the ASIL before decomposition was applied.

Finally, a comment is needed regarding ASIL C. It may have been noted when reading ISO 26262 that ASIL C is something of an anomaly in the scale of rigour for ASIL A through D. According to Part 3, when performing Hazard Analysis and Risk Assessment, the successive categories for “Severity”, “Exposure” and “Controllability” are designed intentionally to be an order of magnitude apart. However, the targets for the “Probabilistic Metric for Hardware Failure (PMHF)” (Part 5, Table 6) show the targets for both ASIL C and ASIL B to be $< 10^{-7}$ h^{-1} (clearly not an order of magnitude apart). Care should therefore be taken when decomposing ASIL C requirements, or decomposing to ASIL B(*x*) requirements.

6 Similar schemes in other standards

IEC 61508 contains a similar scheme to requirements decomposition called “synthesis of elements to achieve the required systematic capability”; so effectively this may be viewed as a composition rather than a decomposition.

The concept of an element having a required “Systematic Capability” of SC *N* expresses the confidence that the element meets the systematic safety integrity requirements of SIL *N* in respect of the safety function(s) allocated to the element.

For two elements each of systematic capability SC *N* (where $N \leq 3$) then it is permitted to claim that the two elements in combination have a systematic capability of (N+1) provided that failure of the safety function is only caused by combined systematic failures of the two elements and that sufficient independence exists between the two elements.

In applying this composition scheme, analysis of common cause failures is required as part of demonstrating the independence of the elements.

Furthermore it is not permitted to apply this composition more than once, i.e. it is not permitted to combine SC *N* elements multiple times in order to achieve a capability of N+2. This restriction is understood to be related to the ability to demonstrate the absence of common cause failures in such cases.

It can be seen that the IEC 61508 scheme is largely comparable to requirements decomposition in ISO 26262 in that:

- It refers only to systematic aspects of safety integrity;
- The independence of (de)composed elements must be shown, including the use of common cause failure analysis.

However key differences compared to the ISO 26262 approach include:

- ISO 26262 permits multiple levels of decomposition, e.g. an ASIL D requirement could be decomposed in two stages into to three requirements inheriting ASIL B(D), A(D) and A(D) across three independent elements; whereas such a (de)composition would not be permitted in the application of IEC 61508;
- Decomposition schemes involving ASIL C (either as the ASIL value of the parent requirement or of one of the decomposed requirements) do not fit the pattern of SC $N \rightarrow$ SC (N–1) + SC (N–1) described in IEC 61508.

7 Correct application of decomposition

This paper has discussed a number of applications where requirements decomposition has been erroneously applied i.e. the approach applied does not correctly meet requirements of ISO 26262 Part 9 Clause 5. In this section some guidance is given which should help in achieving an appropriate application of requirements decomposition.

The Engineer must view requirements decomposition as a requirements manipulation technique and not a design objective. In many cases the Engineer sees that an element needs to implement ASIL D requirements and immediately asks how requirements decomposition can be applied. A better approach is for the Engineer to ask the question “In the context of the architecture, is this requirement a suitable candidate for decomposition?"

Sometimes, designers of a system may be constrained into using a particular component (perhaps as a direct customer requirement); which has not been developed to a level of rigour consistent with the requirements of this application. In this case, requirements decomposition may be a solution if an alternative, redundant and independent means of fulfilling the same safety requirement can be implemented.

If an Engineer decides that the architecture supports decomposition of a requirement, then it is important to ensure that "analysis of dependent failures" (ISO 26262 Part 9 Clause 7) is conducted **as soon as possible**. This analysis should be called for by the impact analysis and change control processes. When considering an application of requirements decomposition it is important not to assume that "it can be done", and instead to ask if it is plausible.

Finally, despite this being allowed by ISO 26262, it is important for the Engineer to think very carefully about applying multiple levels of requirements decomposition. Not only does independence become more difficult to achieve, but care should be taken in that the progressive risk reduction between ASIL levels can be inconsistent (i.e. not always an order of magnitude).

8 Conclusions

Requirements decomposition can be a useful tool in order to practically realize item developments in accordance with ISO 26262. However, it is inappropriate to take an approach of trying to apply requirements decomposition to every and all designs or at all hierarchical levels.

Fundamentally an Engineer (Systems Architect) should ask if the design is suitable for application of requirements decomposition, i.e. can a requirement be independently fulfilled by different elements. If a preliminary answer to this question is in the affirmative, then subject to an analysis of dependent failures requirements decomposition may be a suitable way forward.

Ultimately the architect of the system needs to consider that decomposition should be applied to requirements, and cannot be applied directly to elements. If the requirements decomposition is effective then the ASIL inherited by the element may be reduced which in turn may help in realisation of the product.

Acknowledgements

The authors would like to thank the management of MIRA Limited for permission to publish this paper.

Special thanks are extended to Roger Rivett of JLR for providing Figure 1.

References

[1] ISO 26262: 2011 *Road vehicles – Functional safety*, International Organization for Standardization (2011)

[2] IEC 61508: 2010 *Functional safety of electrical / electronic / programmable electronic safety-related systems*, International Electrotechnical Commission (2010).

[3] D.D. Ward, "The need for safety-related software development standards", *SAE Convergence 2008*, paper number 2008-21-0018 (2008).