The University of Bradford Institutional Repository

Link to publisher version: *https://doi.org/10.4271/2014-01-04*

Citation: Henshall E, Campean IF and Rutter B (2014) A systems approach to the development and use of FMEA in complex automotive applications. SAE International Journal of Materials and Manufacturing. 7(2): 280-290.

2014-01-0740
Published 04/01/2014
Copyright © 2014 SAE International
doi:10.4271/2014-01-0740
saematman.saejournals.org

A Systems Approach to the Development and Use of FMEA in Complex Automotive Applications

Ed Henshall and Ioan Felician Campean
University of Bradford

Brian Rutter
Ford Motor Co.

ABSTRACT

The effective deployment of FMEAs within complex automotive applications faces a number of challenges, including the complexity of the system being analysed, the need to develop a series of coherently linked FMEAs at different levels within the systems hierarchy and across intrinsically interlinked engineering disciplines, and the need for coherent linkage between critical design characteristics cascaded through the systems levels with their counterparts in manufacturing.

The approach presented in this paper to address these challenges is based on a structured Failure Mode Avoidance (FMA) framework which promotes the development of FMEAs within an integrated Systems Engineering approach. The effectiveness of the framework is illustrated through a case study, centred on the development of a diesel exhaust aftertreatment system. This case study demonstrates that the structured FMA framework for function analysis supports an effective decomposition of complex interdisciplinary systems facilitating the DFMEA deployment through a series of containable, structured DFMEAs developed at successive system levels, with clear vertical integration of functional requirements and critical parameters cascade.

The paper also discusses the way in which the approach supports deployment across engineering disciplines and domains, ensuring the integrity of information flow between the design and manufacturing activities.

CITATION: Henshall, E., Campean, I., and Rutter, B., "A Systems Approach to the Development and Use of FMEA in Complex Automotive Applications," *SAE Int. J. Mater. Manf.* 7(2):2014, doi:10.4271/2014-01-0740.

INTRODUCTION

Automotive product development is progressively becoming more complex as systems are required to address both environmental concerns and increasing sophistication of customer expectations, within a highly competitive marketplace. The new technologies introduced in response to these requirements are ubiquitously multidisciplinary, relying on control systems and software to manage the integration of technological systems. Given the ever increasing customer expectations for robust and reliable products, the technical and organisational challenge is to evolve and enhance tools for robust engineering design, in order to cope with the increased technical complexity and multi-disciplinarity of automotive systems.

Failure Modes and Effects Analysis (FMEA) is widely regarded as core tool for evaluating, documenting and mitigating risks in the product creation process, covering both product and manufacturing process design. However, the author's evaluation of the use of the FMEA methodology across a number of OEMs, showed that it is generally not deployed effectively within a systems engineering context. Based on these observations, the main technical challenges of FMEA deployment within a complex automotive system can be summarised as follows:

I. *System Complexity*: The complexity of the system being analysed often results in difficulties in identifying and structuring the functions to be analysed leading to time consuming and resource intensive analysis (1). This usually results in extremely large FMEA documents with a complex unwieldy structure, which includes mixed levels of resolution due to poor definition of the scope in relation to the level of analysis within the systems hierarchy. One consequence of this is that it becomes difficult to evaluate the integrity and comprehensiveness of the document, for example, in ensuring that all critical functions and failure modes have been captured. This makes the document's role of driving both design improvements and efficient verification difficult to achieve (2), and also leads to documents with little reuse value as they are not structured for ease of knowledge storage and recovery.

II. *Systems Deployment*: FMEAs are commonly deployed at the component or subsystem level, once the design is complete, with little evidence that systems integration requirements are consistently captured and addressed (3, 4). While there is recognition that FMEA analysis should be conducted within a systems engineering framework, starting at the "top system" level, FMEA Guidelines (5, 6) give little or no advice on the mechanics of linking between systems levels within a complex system, or the effective deployment and management across engineering teams. Modern automotive systems design typically involves many design teams either within the same company and/or from different suppliers, with each team developing their own FMEA without connecting it effectively with the higher / lower level analysis. It is also the case that different teams, particularly where these are from different suppliers, may employ different approaches to FMEA (7). This makes it extremely difficult to map and manage critical design parameters in an integrated fashion

III. *Multidisciplinarity*: The ubiquitously multidisciplinary nature of modern automotive systems as a mix of electro-mechanical, controls and software introduces a significant challenge stemming from the fact that different engineering disciplines use different approaches to FMEA. In particular software and controls use different approaches compared to conventional electro-mechanical FMEAs which makes validation of the functional integration of the system as a whole a very difficult task (8, 9).

IV. *Multi-domain Integration*: From a systems perspective, Design and Process FMEA are not effectively connected through the systems hierarchy. While there is usually a good connection between component DFMEA and process PFMEA based on critical and significant characteristics (YC/YS and CC/SC) (6), this is not linked to higher level functional requirements and so the effect of the lower level critical/significant characteristic on system performance is poorly understood.

These considerations add to the shortcomings of FMEA stemming from a reliance on brainstorming in conducting function analysis, functional decomposition and cause identification as discussed previously (10). Even if expert facilitation of both the FMEA process itself and in brainstorming is employed as recommended (2, 11), this is unlikely to systematically tackle the complexity challenge. Considerable attention has also been paid in the literature to difficulties associated with the use of the use Severity, Occurrence and Detection rating scales within the FMEA methodology with numerous recommendations for improvement being made (3). While it is important that any use of rating scales needs to be meaningful the present authors consider that a good understanding of the function and hence failure modes associated a particular system design and their decomposition is more fundamental.

The fundamental role of the FMEA in effective Failure Mode Avoidance (FMA) has been widely discussed, most prominently by Davis (12) who regarded the FMEA as the "superordinate document" for FMA. This reflects Davis' view that the FMEA is best placed to comprehensively document the organization's knowledge on failure mode avoidance, covering failure modes, effective detection events for failure modes, design guidelines and robustness countermeasures to avoid failure modes due to mistakes and lack of robustness, and robust design verification plans and outcomes. The linkage and integration between FMEA and robustness tools has been discussed by several other authors(6, 13).

The main contribution of the Failure Mode Avoidance process proposed by the research team at the University of Bradford Engineering Quality Improvement Centre (BEQIC) consists of the introduction of a structured framework for function analysis of complex engineering systems (10), which facilitates complex system decomposition on a functional basis, providing an effective approach for task breakdown for the conventional deployment of failure avoidance through DFMEA with robustness linkages. This structured approach underpinned by fundamental systems thinking in identification of the functional chains significantly enhances the rigour of the analysis, removing the brainstorming basis of conventional function analysis feeding DFMEA development.

The introduction of the 4-step FMA process of Function Analysis, Function Failure Analysis, Robust Countermeasure Development and Robust Design Verification (14) further strengthened the information flow between the engineering tools. Several industrial case studies have been used to demonstrate the effectiveness of this framework, including the way in which information flow within the FMA process enhances the development and integrity of the FMEA (10, 14, 15). The strategy for "systems engineering based on failure mode avoidance" has been discussed in (15) based on the horizontal deployment of the FMA process at each system level with vertical integration between the systems levels, illustrated with a case study on a Diesel aftertreatment design analysis. In this paper we reconsider this case study as the basis for demonstrating the effectiveness of the FMA process to support the effective deployment of the FMEA within the systems engineering framework.

The aim of the work presented in this paper is to consider how the FMA process and tools support the effective deployment of FMEAs within an integrated systems engineering approach, addressing the challenges outlined above effectively. Key aspects in terms of the novelty in the approach rest on the identification and organization of functions through the analysis of state flow and the use of detailed interface analysis to identify causes of functional failure.

DFMEA DEVELOPMENT AND USE WITHIN A SYSTEMS ENGINEERING FRAMEWORK

DFMEA Development within the FMA Process

Within the FMA process, *functional analysis*, in which function identification and decomposition is achieved using fundamental engineering thinking through the development of a System State Flow Diagram (10) is a key tool for the decomposition of the complex system. Based on the System State Flow Diagram a Function Tree and a Boundary Diagram, as a schematic representation of the design structure that delivers the system functionality, can be derived. The Boundary Diagram is then used as the basis for the *interface analysis* in which the exchanges of energy, material and information which occur both between the design elements that comprise a system and between the design elements within the system and elements external to the system are identified in an Interface Matrix. Interface Analysis Tables are used to extend the information contained within the Interface Matrix, by documenting the nature of the interface exchange and the functional requirements associated with each exchange. Figure 1 illustrates the information flow between the different FMA tools within Function Analysis.

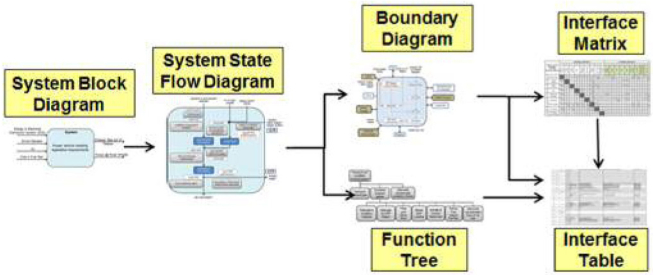
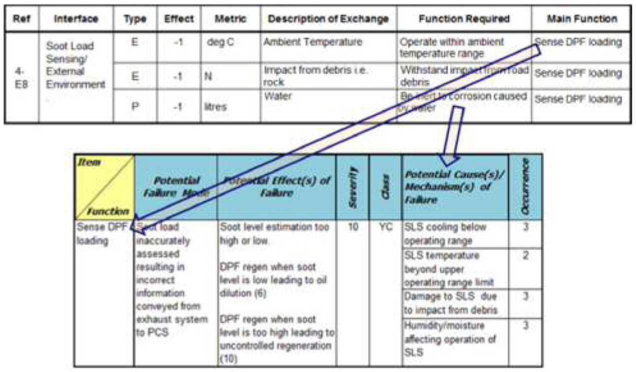


Figure 1. Information Flow within Function Analysis *(For larger version of graphics see Appendix)*

The Interface Table includes a description of each exchange that occurs at the interface being analysed along with the interface function required to manage that exchange. The main function of the exhaust system, which can be affected by the interface exchanges, is identified from the Function Tree and documented in the table.

Thus, the Interface Analysis Table performs a very important role in documenting all the functional requirements for system integration, covering both the main functions and the interface functions. This information is also fundamental for the development of the DFMEA, which aims to identify and evaluate system function failure modes in terms of their effects and root causes. The mechanics of the link between the interface table and DFMEA are illustrated in Figure 2. Consistent with the conventional functional based FMEA methodology (5), the main functions in the DFMEA can be extracted from the Interface Table. Each of the four types of functional failure mode of the main function (no function, partial function, intermittent function and unintended function) are considered in turn in order to establish their effects and causes. Effects are identified in the usual manner (5) along with the severity rating and class of each effect.



Ref	Interface	Type	Effect	Metric	Description of Exchange	Function Required	Main Function
4-E8	Soot Load Sensing/ External Environment	E	-1	deg C	Ambient Temperature	Operate within ambient temperature range	Sense DPF loading
		E	-1	N	Impact from debris i.e. rock	Withstand impact from road debris	Sense DPF loading
		P	-1	litres	Water	Be inert to corrosion caused by water	Sense DPF loading

Item / Function	Potential Failure Mode	Potential Effect(s) of Failure	Severity	Class	Potential Cause(s)/ Mechanism(s) of Failure	Occurrence
Sense DPF loading	Soot load inaccurately assessed resulting in incorrect information conveyed from exhaust system to PCS	Soot level estimation too high or low DPF regen when soot level is low leading to oil dilution (6) DPF regen when soot level is too high leading to uncontrolled regeneration (10)	10	YC	SLS cooling below operating range	3
					SLS temperature beyond upper operating range limit	2
					Damage to SLS due to impact from debris	3
					Humidity/moisture affecting operation of SLS	3

Figure 2. Transfer of Information from Interface Table to DFMEA

Since interfaces exchanges affect the performance of the system, the failure of an interface function, specified to manage such an exchange, will potentially cause system failure; i.e. failure modes associated with interface exchange functions are potential causes of system main function failure modes. This means that the Interface Table, by providing a comprehensive source of information on interface exchanges and the functions required to manage them, furnishes information on the causes of failure of main functions. Information is transferred from the Interface Table to the DFMEA in terms of potential causes of failure through answering the questions "How can this exchange cause failure of the main function?" and/or "Which failure modes of the interface function are causes of failure of the main function?". The identification of potential causes of failure is then repeated for each main function in turn.

This demonstrates that the DFMEA is substantially based upon information contained in the Interface Table in terms of both structure and content. The comprehensive nature of the Interface Table means that all main function and associated failure modes of a system should be analysed and that a comprehensive list of causes based on the identification of all interface functions are identified and documented in the DFMEA. Since the Interface Table is based on the use of a Boundary Diagram developed from a System State Flow Diagram (15) which itself is established on the basis of fundamental engineering thinking this ensures that the DFMEA is self-contained by being limited to a review of (only) the main functions that achieve system function. The structure inherent in the Interface Table in terms of the listing of Main Functions and Interface Functions is passed across to the DFMEA meaning that the related information in the two documents can be easily reconciled.

DFMEA Development across Engineering Disciplines

A strong feature of the SSFD based function analysis and decomposition is that the functional analysis framework (and by implication the FMA process) can be deployed across engineering disciplines. This provides the basis for a common approach to both hardware and software DFMEA development. For example, the DFMEA for the software Control Feature that manages regeneration of the Diesel Particulate Filter was developed by first conducting interface analysis on the basis of a Boundary Diagram which is derived directly from the System State Flow Diagram for the feature (Figure 3) (15).

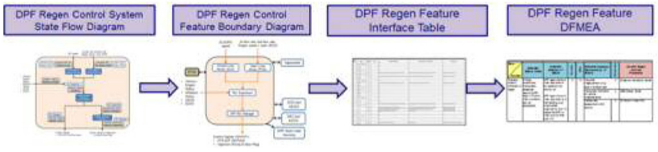


Figure 3. Information Flow for DPF Regeneration Feature Analysis (SSFD through Interface Analysis to DFMEA)

The nature of the System State Flow Diagram ensures that the Function Analysis process and hence the DFMEA which results from this analysis is effectively maintained at a single level within the systems hierarchy, with clear linkage to the other levels. This latter aspect of the DFMEA development process is considered next.

DFMEA as a Part of Systems Engineering

DFMEA Systems Cascade

Systems engineering is facilitated by conducting the 4-Step FMA process at each level of the systems hierarchy (15). DFMEAs are developed at each level using the process described above, with the System DFMEA being based on system level interface analysis and the Subsystem DFMEA being based on subsystem level interface analysis, etc. This ensures that the content of a DFMEA at a particular level reflects that level rather than having a mix of content reflecting multiple levels. An FMA approach to systems engineering is founded on the cascade and deployment of interface exchanges through the systems hierarchy with interface analysis forming the central core or backbone of the process (15). Thus interface analysis provides the conduit for the transfer of coherent information between DFMEAs for the same system at the different system levels, as shown in Figure 4.

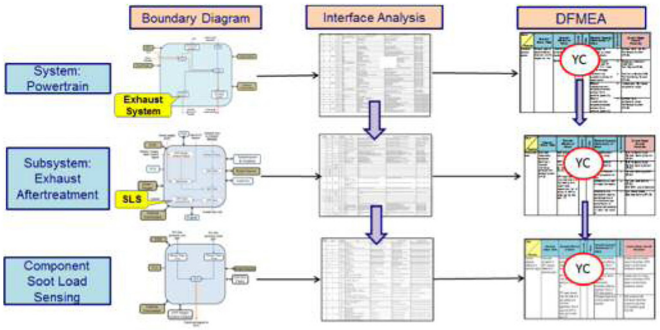


Figure 4. Systems Engineering Cascade and Information Flow to DFMEAs

As illustrated is in Figure 4 by the Boundary Diagrams the Exhaust Aftertreatment system is "nested" within the Powertrain system; similarly, the DPF Soot Load Sensing (SLS) is in turn nested within the Exhaust Aftertreatment system. This relationship is reflected in the formal mechanism for the cascade of external interfaces within the Systems Engineering framework described in (15). DFMEAs are developed at each level of the systems hierarchy based on the Interface Analysis conducted at that level. Thus a coherent and contained series of DFMEAs are developed at each level with the DFMEAs at each level linked through the interface cascade to the level below/above through the functional requirements cascade which is a feature of the interface analysis cascade. Considering this linkage bottom-up, the SLS Component Level DFMEA is linked in the Exhaust Subsystem DFMEA which in turn it is linked into the System Level Powertrain DFMEA. This linkage to the higher/lower levels is true for all DFMEAs at any particular level. The coherent relationship between DFMEAs at different levels is also seen in the cascade of critical characteristics or YCs (5) also marked in Figure 4. By way of example the critical characteristic associated with meeting the emission requirements at system level is coherently linked through the cascade of DFMEAs to, amongst other things, a critical characteristic associated with the conversion of the pressure across the DPF into a voltage signal by Soot Load Sensing subassembly.

The Boundary Diagrams, and hence more fundamentally, the System State Flow Diagrams at each of the system levels, structures the cascade and substantiates the linkage of DFMEAs within the systems framework. A complete DFMEA cascade is illustrated in Figure 5 with the Powertrain System DFMEA being cascaded/linked to a series of Subsystem DFMEAs corresponding to the design elements that achieve the Powertrain main functions as defined on the Powertrain System State Flow Diagram. Each Subsystem DFMEA is then cascaded/linked to a series of Component DFMEAs which correspond to the design elements that comprise each particular subsystem; the cascade for the Exhaust Subsystem being illustrated in Figure 5. Figure 5 is somewhat simplified by not including all of the design elements that typically make up a Powertrain and Exhaust.

.

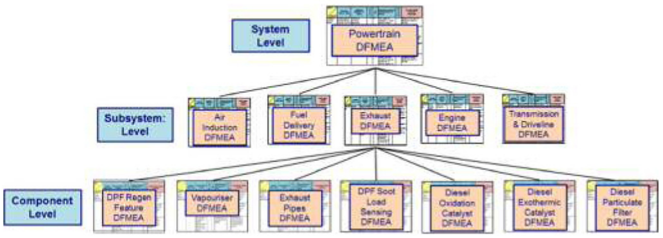


Figure 5. DFMEA Cascade between the Systems Levels

A practical benefit of using the structured FMA process to facilitate the development of DFMEAs within a systems engineering framework is that it replaces the often long and unwieldy DFMEA documents with a series of comprehensive but contained DFMEAs of a few pages length, with coherent structural and information links between them.

DFMEA, Fault Tree Analysis and the P-Diagram

The structural weakness of DFMEA is that the principal information flow is horizontally across the document making the identification of failure modes due to a combination of causes difficult to identify. Fault Tree Analysis is widely used as a structured approach for identifying combinations of events as causes for system failures. Function Fault Tree Analysis is (FFTA) (16) underpinned by a function based approach to constructing a fault tree, based on the premise that a fault tree can be regarded as the mirror image in the failure domain of the corresponding function tree. Therefore, the FFTA development can be fully integrated with the FMA function analysis tools. In particular the higher levels of a Function Fault Tree are a mirror in the failure domain of the associated Function Tree set in the success domain while lower level causal events on the tree are based on information flow from the Interface Table.

From the point of view of function failure analysis, the main advantage of the FFTA is that it is based on a vertical information flow, developed top-down through the successive systems engineering levels. On this basis the FFTA can be used as a complementary tool for the DFMEA to overcome its weakness in identifying failure modes causes as combination of events. Potential combinations of causes of failure of the top event can be readily seen on the Function Fault Tree because of its graphical nature and since the causal events on the DFMEA and Function Fault Tree are both identified on the basis of the same interface analysis these combined causes of failure can be documented directly as potential causes of failure of the main function on the DFMEA. That is, the DFMEA can be updated to include combinations of causes of failure. Figure 6 illustrates the information flow between tools, noting that information from the upper events on the FFTA is used to update the DPF Regen Feature DFMEA, while information from the lower events on the Fault Tree is used to update the SLS DFMEA. Because of its nature the Feature may be considered to be at an intermediate level between the Exhaust Aftertreatment System at subsystem level and the SLS at component level.

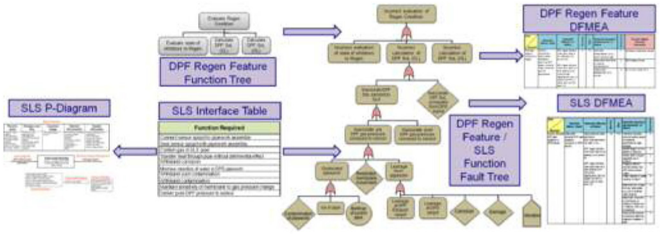


Figure 6. Information Flow Interface Table to P-Diagram and DFMEA via Function Fault Tree

Having identified a comprehensive list of potential causes of failure on the DFMEA the existing countermeasures are assessed as Current Prevention Controls and where these are deemed inadequate, enhanced countermeasures are developed and documented on the DFMEA as Recommended Actions. Together the P-Diagram and Robustness Checklist (RCL) (6) along with the FFTA facilitate the establishment of appropriate countermeasures. FFTA assists the identification of countermeasures because it focuses attention on combinations of causes that might otherwise have been missed and conversely on functions that need to be achieved (16). In terms of this latter point FFTA is particular powerful at identifying countermeasures based on functional redundancy. The P-Diagram and RCL facilitate the identification of countermeasure by firstly identifying noise factors as causes of robustness failure and subsequently aiding the establishment of appropriate noise factor management strategies. Since noise factors can only affect system performance by an exchange of energy, material or information at an interface the Interface Table provides comprehensive information on noise factors i.e. the i.e. interface analysis facilities the development of the P-Diagram (as illustrated in figure 6) and hence RCL (10).

While the top event on the Function Fault Tree can be at any level with the tree extending down to root causes at component, level in the authors' experience P-Diagrams are most effective when developed at the lower system levels of analysis. This is because system complexity at the higher system levels along with the large number of noise factors affecting the system is such that it becomes very difficult to isolate the effect of individual noise factors on individual components. The need to isolate the effect of individual noise factors on individual components is necessary since it is only at component level that the design actions required to implement countermeasures to robustness failures are implemented. Although countermeasures are implemented at the lower (component) levels of the systems hierarchy the need for countermeasures is identified at the higher levels for those aspects of system or subsystem which are known not to be robust. Thus, typically, the System and Subsystem level DFMEA Recommended Actions reflect plans to implement countermeasures at Component level while the Component level DFMEA Recommended Actions reflect the detail of the countermeasures.

DFMEA and Design Verification

Having implemented countermeasures documented as Recommended Actions in the DFMEA there is a need to verify that each countermeasure is effective in ensuring that the failure mode with which it is associated does not occur. The Robustness Checklist facilitates the incorporation of each significant noise factor effect into a least one of the series of tests that make up the Design Verification Plan (DVP) in a way that minimises the total number of tests that are required (17).

The DFMEA is then updated with the results of design verification being documented as the Results of Recommended Actions.

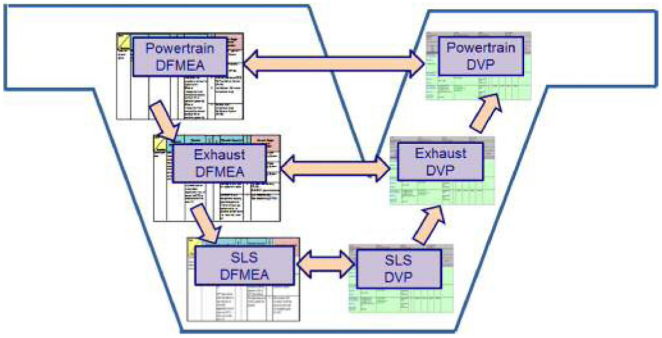


Figure7. Information flow between DFMEA and DVP

As depicted in Figure 7 the system level DFMEA facilitates the development of the system level design verification testing documented in the system level DVP with the subsystem and component level DFMEAs facilitating the development of the subsystem and component level verification testing respectively. The functional decomposition depicted in requirements cascade through the DFMEAs and achieved through *function analysis* and *interface analysis* support the upward integration and linkage of the different levels of design verification.

Process FMEA and YC to CC Linkage

The process outlined in this paper for deploying DFMEAs within a systems engineering context can be extended to the process domain and the development of PFMEAs. The correspondence of the function analysis tools deployed in the design domain with similar tools used in the manufacturing process domain has been discussed in (18) on the basis of an aerospace manufacturing process. Similarly, the Dynamic Control Planning methodology (19) used in the automotive industry can be shown to be directly analogous to the function analysis process discussed in this paper, thus enabling coherent information flow between the DFMEAs developed during product development and PFMEAs developed during manufacturing process development. This enables, for example, the cascade of YC/YS critical/significant characteristics through the system hierarchy to be coherently linked to CC/SC critical/significant manufacturing characteristics (5) which are then documented as such on the associated PFMEA.

DISCUSSION

This paper introduced a Systems Engineering framework based on a Failure Mode Avoidance process, within which DFMEAs can be developed in an integrated, rigorous and structured manner The DFMEA is created as an integral part of a structured 4-Step FMA process based on coherent information flow between engineering tools both within, and between, the different levels of the systems hierarchy. The way in which the use of the BEQIC 4-Step FMA process addresses the challenges to effective deployment of DFMEA within a System Engineering framework is discussed next.

Challenge I: System Complexity

The challenge associated with System Complexity is to structure the DFMEAs with clearly defined functions such that all the analysis contained within any one document is at the same level of resolution in terms of the systems hierarchy. *Function Analysis* enables both the main functions that convert the inputs to outputs in a system and the functions that manage the interface exchanges that affect system performance to be identified on the basis of structured fundamental engineering thinking. This means that each main function which is the subject of DFMEA analysis are pre-defined in an Interface Table before DFMEA development commences with all the main functions being at the same level of resolution in terms of the systems hierarchy. Equally since the interface functions contained within an Interface Table will also be at the same level the causal analysis within the corresponded DFMEA will also be at level of resolution which is appropriate to the level of DFMEA being developed. The manner in which the interface table is developed also leads to the development of a contained FMEA document with straightforward access to the key knowledge that it contains both in terms of ability to being able to assess the integrity of that knowledge and its subsequent use.

This is a significant improvement over the conventional approach to DFMEA development in which the functions to be analysed may be identified either through discussion alone or on the basis of a Function Tree which itself is developed on the basis of brainstorming. The authors' experience of using the conventional approach is that, in starting a DFMEA a considerable amount of time is expended in gaining agreement within the team developing the DFMEA as to what functions should be analysed with the resulting discussion tending to identify functions at different levels of the systems hierarchy; this is particularly true when different people in the team have design responsibility for particular aspects of a system since they tend to focus more intently on “their” part. The result of this conventional process is a list of main functions which is not necessarily exhaustive (i.e. some functions are missed) and functions which correspond to different levels of the systems hierarchy leading to a lengthy DFMEA document of doubtful integrity which is difficult to analyse.

Challenge II: Systems Deployment

The Systems Deployment challenge is to develop a series of contained DFMEAs which link coherently across and between levels of the systems hierarchy. By decomposing the system on a structured functional basis, the deployment of the DFMEA within the system can be managed through a series of “nested” DFMEAs, matching the nested structure of the system decomposition, as illustrated in Figure 4. This has a great benefit for the deployment, in that the effort of the FMEA teams is much more focused, hence more manageable from the time point of view; a typical DFMEA at any level in the hierarchy (illustrated in Figure 5) can be completed with hours of team effort.

This addresses a common weakness of current practice which often leads to the production of large DFMEAs, sometimes extending to several hundreds of pages in length which prove extremely difficult to analyse particularly in terms of the linkages between system levels (15). Additionally within the conventional approach lower level Component DFMEAs are often developed in isolation of, and hence with no linkage to, higher level systems analysis.

Challenge III: Multidisciplinarity

The challenge associated with multidiciplinarity is the ability to use a common approach to the development of DFMEAs across the different engineering disciplines. The BEQIC FMA process is applicable across all engineering disciplines so enabling the development of DFMEAs for each separate discipline to be based on the same fundamental processes. This contrasts to the current situation where typically different approaches are used in the development of DFMEAs in different disciplines. Equally important, using a common approach across engineering disciplines means that a DFMEA can document potential causes of failure associated with poor systems integration. For example, it can highlight potential causes of failure at the interfaces between of a software based Control Feature and the hardware that the feature manages.

Challenge IV: Multi-Domain Integration

The multi-domain challenge is to link the coherent information flow through the system hierarchy related to critical and significant characteristics in design (YC/YS) to their counterparts of critical/significant characteristics in manufacturing (CC/SC).

The coherent linkage that the 4-Step FMA process provides between YC characteristics through the systems hierarchy and on to CC characteristics in manufacturing can be compared to the conventional approach in which, in the absence of consistent linkage between DFMEAs at the different levels, there is a tendency to assign a product design characteristic as “critical” within a Component DFMEA alone. This absence of coherent linkage between a Component DFMEA and higher level DFMEAs through interface analysis means that the effect of any particular component level design characteristic on system performance may not be fully understood and hence the true significance of the characteristic is not fully evaluated. This means that, in the absence of knowledge, characteristics are sometimes deemed to be significant “to be on the safe side” which results in an unnecessary level of manufacturing control and hence unnecessary cost. Equally some critical characteristics may not be identified if the significance of characteristics is judged through component level DFMEAs alone which may result in costly failures.

Other Benefits

Within the 4-Step FMA process potential causes of failure of each main function are identified as failure events associated with the functions required to manage interface exchanges that have been identified as a part of Function Analysis. Evidence from the field demonstrates that the majority of failures of automotive systems are due to poor management of interface exchanges (20) emphasising the importance of identifying such causes during product design. Indeed it can be argued that the vast majority, if not all, causes of failure are due to interface exchanges meaning that developing a DFMEA within the 4-Step FMA process results in an exhaustive list of causes being identified. This can be contrasted with the conventional approach to the identification of causes on a DFMEA through brainstorming which as the experience from industry practice indicates, results in a significant number of potential root causes not being identified during the product design process and hence effective countermeasures not being established. Additionally it is the authors' experience that in identifying functions (and hence and function failure modes) through brainstorming that team members often confuse failure modes and causes thereby documenting some of these in inappropriate columns on the DFMEA leading to a loss of structure of the document. The 4-Step FMA process helps to overcome this confusion by identifying main functions and information on the causes of failure of these main functions in clearly separated columns in the Interface Table.

The inclusion of the graphical tool of Function Fault Tree Analysis within the BEQIC approach to the development of DFMEAs, with the DFMEA and associated Function Fault Tree both being developed on the basis of information flow from common interface analysis, means that the DFMEA can be easily updated to include multiple cause events where appropriate. Like the DFMEA and Function Fault Tree an associated P-diagram is also developed on the basis of the common interface analysis information with the P-Diagram also benefiting from being a graphically based document. The P-Diagram facilitates the flow of information, via the RCL, that enables the development of effective Design Verification. In convention usage, for example, as a part of the Ford Robustness Linkages process (6) the P-Diagram identifies noise factors as causes of failure to be included in the DFMEA. However, since the conventional approach to the development of P-Diagrams involves brainstorming noise factors this means that the causes subsequently documented in the DFMEA are based on brainstorming. In the 4-Step FMA process the use of the P-Diagram is reserved to help develop the countermeasures subsequently documented on the DFMEA with the noise factor being identified on the basis of interface analysis which is itself based on fundamental engineering thinking. The P-Diagram's role of identifying causes of failure on the DFMEA is seen as redundant since both these causes and the noise factors identified on the P-Diagram are sourced from common interface analysis.

As a final point the authors' experience of developing DFMEAs using the 4-Step BEQIC FMA process within a systems engineering framework has been encouraging in that more contained and hence manageable set of coherently related documents covering all levels of the systems hierarchy have been generated with the overall set of DFMEAs being significantly more comprehensive than the previous generation of DFMEAs.

REFERENCES

1. Hunt J. E., Pugh, D. R., Price, C. J., Failure mode effects analysis: a practical application of functional modelling. Applied Artificial Intelligence, Vol 9 Issue 1, pp33-44, 1995.
2. Carlson, C., Lessons Learned for Effective FMEAs, 58th Annual Reliability and Maintainability Symposium (RAMS), IEEE, 2012.
3. Liu, Hu-Chen, Liu, Long, Liu, Nan, Risk evaluation approaches in failure mode and effects analysis: A literature review, Expert Systems with Applications, Vol 40, pp 828-38, 2013.
4. Kumar, C., Maass, E., FMEAs Using a Functional Modeling Based Approach, 59th Annual Reliability and Maintainability Symposium (RAMS), IEEE, 2013.
5. Potential Failure Mode and Effect Analysis (FMEA), 4th Edition, AIAG, 2008.
6. Failure Mode and Effects Analysis FMEA Handbook (with Robustness Linkages) v4.2, Ford Motor Company, 2011.
7. Fritzsche, R., “Failure Mode and Effects Analysis (FMEA): A Comparison Between VDA-Approach Versus QS-9000,” SAE Technical Paper 2011-01-2280, 2011, doi:10.4271/2011-01-2280.
8. Ozarin, N., Bridging Software and Hardware FMEA in Complex Systems, 59th Annual Reliability and Maintainability Symposium (RAMS), IEEE, 2013.
9. Raheja, D., “Software FMEA: A Missing Link in Design for Robustness,” SAE Technical Paper 2005-01-0817, 2005, doi:10.4271/2005-01-0817.
10. Campean, I., Henshall, E., Brunson, D., Day, A. et al., “A Structured Approach for Function Analysis of Complex Automotive Systems,” *SAE Int. J. Mater. Manuf.* 4(1):1255-1267, 2011, doi:10.4271/2011-01-1268.
11. Bluvband, Z., Grabov P., Failure Analysis of FMEA, 55th Annual Reliability and Maintainability Symposium (RAMS), IEEE, 2009.
12. Davis, T., Failure Mode Avoidance Seminar, University of Bradford, UK, November 2007.
13. Kumar, C., Maass, E., FMEAs Using a Functional Modeling Based Approach, 59th Annual Reliability and Maintainability Symposium (RAMS), IEEE, 2013.
14. Campean, I., Henshall, E., Systems Engineering Design through Failure Mode Avoidance - an Automotive Industry Perspective, in Enduring and Cost-Effective Engineering Support Solution, Eds Roy et al, Proc 1st Int Conf on Through-life Engineering Services Conference, Cranfield University Press, 99-107, 2012.
15. Campean, I., Henshall, E., and Rutter, B., “Systems Engineering Excellence Through Design: An Integrated Approach Based on Failure Mode Avoidance,” *SAE Int. J. Mater. Manf.* 6(3):389-401, 2013, doi:10.4271/2013-01-0595.
16. Campean, F. and Henshall, E., “A Function Failure Approach to Fault Tree Analysis for Automotive Systems,” SAE Technical Paper 2008-01-0846, 2008, doi:10.4271/2008-01-0846.
17. Henshall, E. and Campean, F., “Design Verification as a Key Deliverable of Function Failure Avoidance,” *SAE Int. J. Mater. Manuf.* 3(1):445-453, 2010, doi:10.4271/2010-01-0708.
18. Goodland, J, Campean IF, Caunce, A, Victory, J, Jupp, M, A Manufacturing Failure Mode Avoidance for Aerospace Manufacturing, IMECE2013-64606, Proceedings of the ASME 2013 International Mechanical Engineering Congress and Exposition.
19. Ford Motor Company, The Strategy of Dynamic Control Planning, Training and Reference Manual, 1997.
20. Webb, R.D. Investigation into the application of robustness and reliability tools to the design process. MSc thesis, University of Bradford, 2002.

APPENDIX

LARGER SCALE VERSION OF GRAPHICS

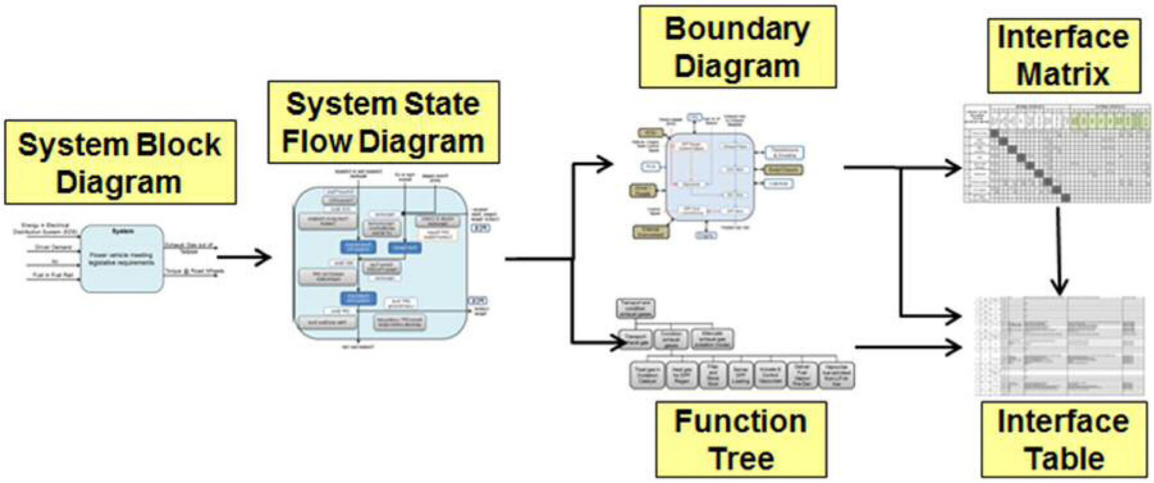
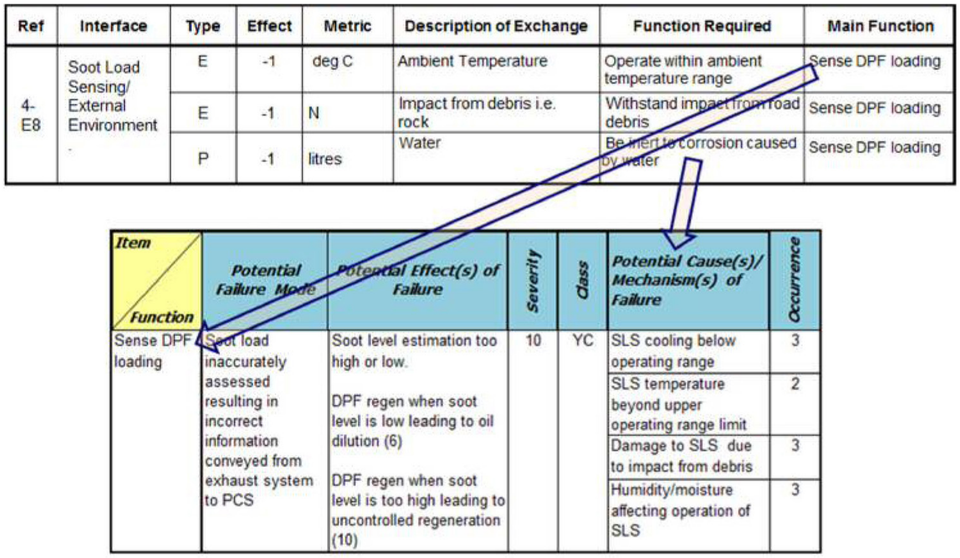


Figure 1. Information Flow within Function Analysis



Ref	Interface	Type	Effect	Metric	Description of Exchange	Function Required	Main Function
4-E8	Soot Load Sensing/ External Environment	E	-1	deg C	Ambient Temperature	Operate within ambient temperature range	Sense DPF loading
		E	-1	N	Impact from debris i.e. rock	Withstand impact from road debris	Sense DPF loading
		P	-1	litres	Water	Be inert to corrosion caused by water	Sense DPF loading

Item / Function	Potential Failure Mode	Potential Effect(s) of Failure	Severity	Class	Potential Cause(s)/ Mechanism(s) of Failure	Occurrence
Sense DPF loading	Soot load inaccurately assessed resulting in incorrect information conveyed from exhaust system to PCS	Soot level estimation too high or low. DPF regen when soot level is low leading to oil dilution (6) DPF regen when soot level is too high leading to uncontrolled regeneration (10)	10	YC	SLS cooling below operating range	3
					SLS temperature beyond upper operating range limit	2
					Damage to SLS due to impact from debris	3
					Humidity/moisture affecting operation of SLS	3

Figure 2. Transfer of Information from Interface Table to DFMEA

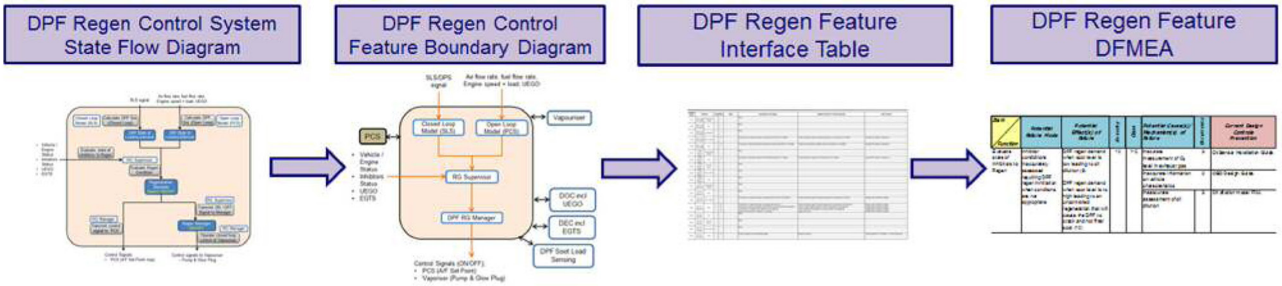


Figure 3. Information Flow for DPF Regeneration Feature Analysis (SSFD through Interface Analysis to DFMEA)

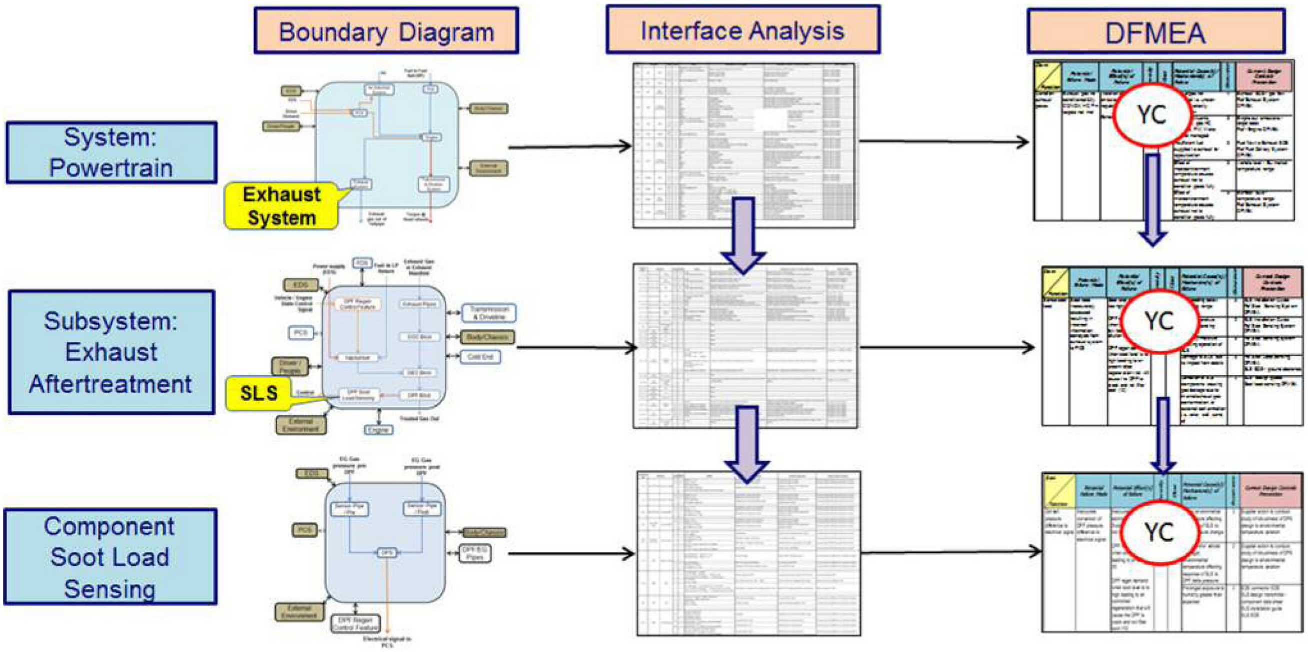


Figure 4. Systems Engineering Cascade and Information Flow to DFMEAs

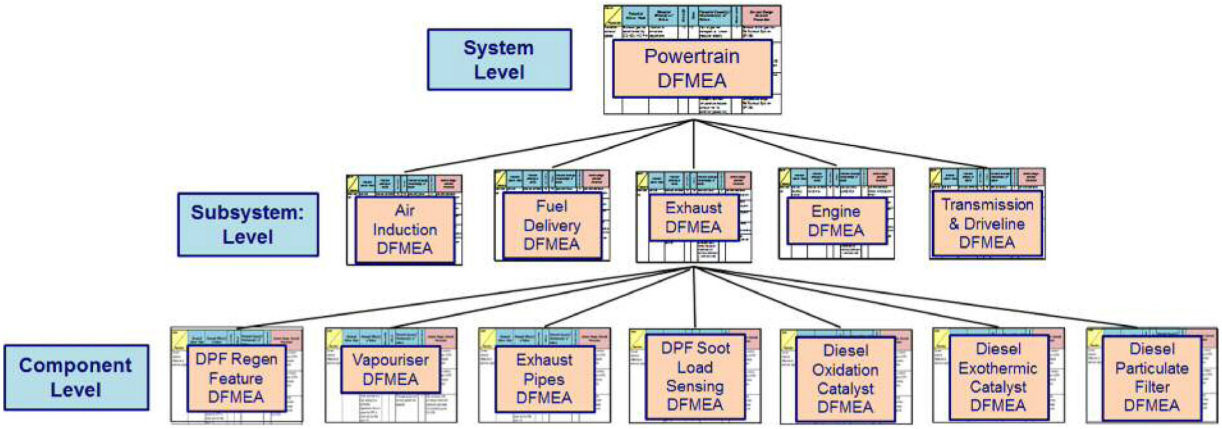
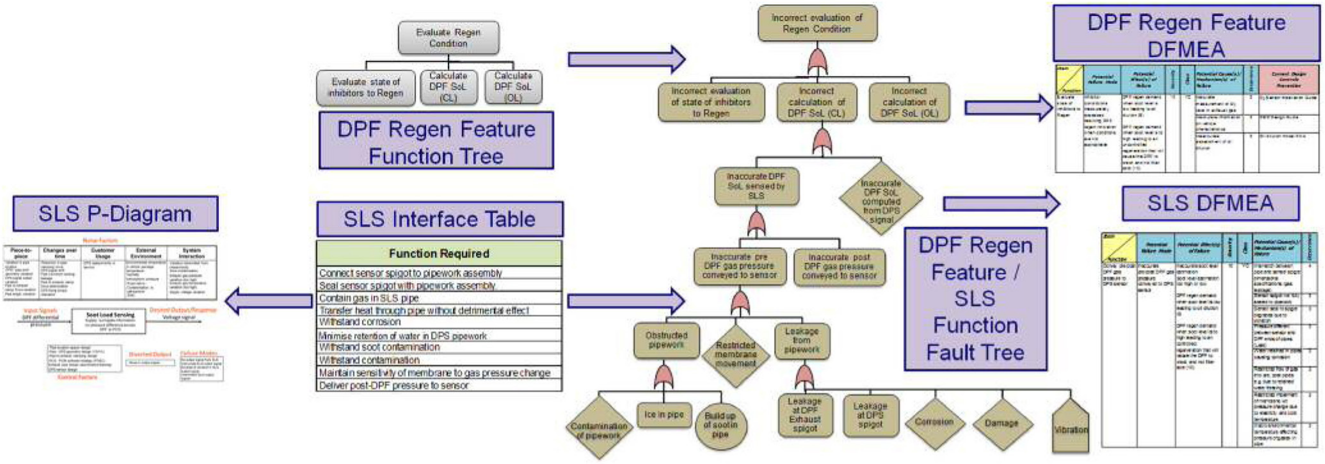


Figure 5. DFMEA Cascade between the Systems Levels



Function Required
Connect sensor spigot to pipework assembly
Seal sensor spigot with pipework assembly
Contain gas in SLS pipe
Transfer heat through pipe without detrimental effect
Withstand corrosion
Minimise retention of water in DPS pipework
Withstand soot contamination
Withstand contamination
Maintain sensitivity of membrane to gas pressure change
Deliver post-DPF pressure to sensor

Figure 6. Information Flow Interface Table to P-Diagram and DFMEA via Function Fault Tree

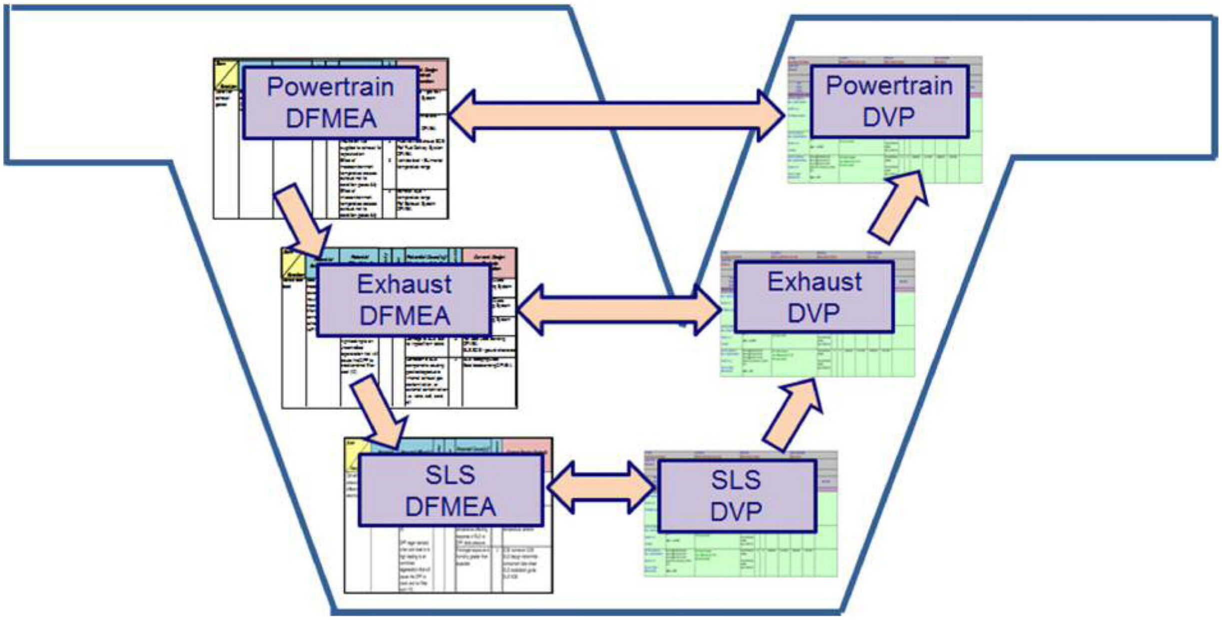


Figure7. Information flow between DFMEA and DVP