Reusing FIDES knowledge in the MéDISIS Dysfunctional Behavior Database

Robin Cressent, Vincent Idasiak, Frédéric Kratz

Reusing FIDES knowledge in the MéDISIS Dysfunctional Behavior Database

Robin Cressent[a*], Vincent Idasiak[a], Frederic Kratz[a]
[a]PRISME / ENSI de Bourges, Bourges, France

Abstract: For 3 years our research team has applied on different industrial projects our MéDISIS methodology. The MéDISIS methodology aims to ease the integration of dependability analysis through the system engineering process. MéDISIS represents now a complete framework described through various publications. Currently, the use of MéDISIS relies on a functional system model in SysML, the Dysfunctional Behavior Database and processes of translation from SysML to FMEA, Altarica DataFlow, AADL and Simulink to perform dependability studies. Until recently, the DBD was a tool dedicated to MéDISIS. Consequently, the DBD was only designed to be updated through dependability analysis of project implementing the MéDISIS methodology. We describe in this article how we can complete the DBD with knowledge from other sources such as reliability repository. The main source we have studied is the FIDES guide. We highlight how the meta-model of our DBD and the model of the dependability database allows to connect them. We describe the process leading to failure law definition and failure rate calculation in the FIDES guide. It matches perfectly several concepts that are associated to the description of failure modes in the DBD meta-model.

Keywords: Model-Based System Engineering; SysML; FIDES guide; Reliability database.

1. INTRODUCTION

In the current highly competitive context, designing and producing systems needs to rely more and more on optimized processes to master complexity in a restrained time, while validating system performances. In every project but particularly when designing critical systems, dependability analysis needs to be integrated in the design process to benefit from it since the early specification phases. Moreover it has been proven that it was cost-effective to detect potential failures as soon as possible, since it permits to avoid reengineering costs [1-3].

Design processes exchanging data between system engineering activities and dependability studies were described in [4] and in various industrial dependability standards [2,3]. Our work also described various tools to ease the exchanges between system engineers and dependability experts. These tools are organized in a framework supporting the MéDISIS methodology. It relies on the Model-Based System Engineering (MBSE) [5] approach and the use of SysML (System Modeling Language) [6] as the modeling language for system engineering activities. MéDISIS then provides means of extracting information from SysML models to perform dependability studies. Currently, MéDISIS counts four processes that translate SysML models into target languages: the FMEA generation process and the Altarica DF translation process were described in [7], the SysML to AADL translation process and the process to Simulink were described in [8,9].

To support this translation processes the framework also embeds the Dysfunctional Behavior Database (DBD). The DBD grants the possibility to import and perpetuate feedback gained on previous studies concerning dependability features of components. The DBD is necessary because system engineering activities don’t deal with dysfunctional behavior.

Traditional reliability databases are necessary to manage dependability aspects during a project to perform reliability analysis and match failure rate requirements. Reliability databases evolved since the first dependability information collection: “the Martin Titan Handbook” [10]. This handbook consisted in gathering failure rates for electrical and mechanical component in a uniform way used during a large missile design project. This evolution transcribed by [11] explains how upon time reliability databases took the benefits of supporting computer systems and then introduced the environmental conditions and the reasons of failure in failure rates computation.

Yet, the current databases are not already formed to fully integrate MBSE processes. In fact their format is not model-oriented, forcing a manual application of their concepts which point out the need for new concepts of reliability and risk analysis databases.

Such a reliability database would be of great help to support the Model-Based Safety Assessment activities that currently begin to widely spread amongst the safety analysts, see for example [12,13]. The models contained in these databases are expected to help conducting fault injection analyses and failure logic modeling that requires the faulty modeling of components and its failure behavior.

We can note that the first version of Dysfunctional Behavior Database (DBD) we gave in [7] is a valuable starting point regarding its structure and the predicted help it will provide. However, we noticed that with each new step in the evolution of the reliability databases, the field of knowledge stored increased. That fact implies that newer databases must embed data from former ones. Extending this reasoning to our case, our DBD needs to embed data from current reliability databases such as FIDES [14] or Mil-HBDK 217f [15].

In this article we will first present shortly our DBD and its meta-model. Then we will study the organization of knowledge stored in the FIDES reliability database. Ultimately, we will highlight the connections between the concept of FIDES and our DBD.

2. THE DYSFUNCTIONAL BEHAVIOR DATABASE

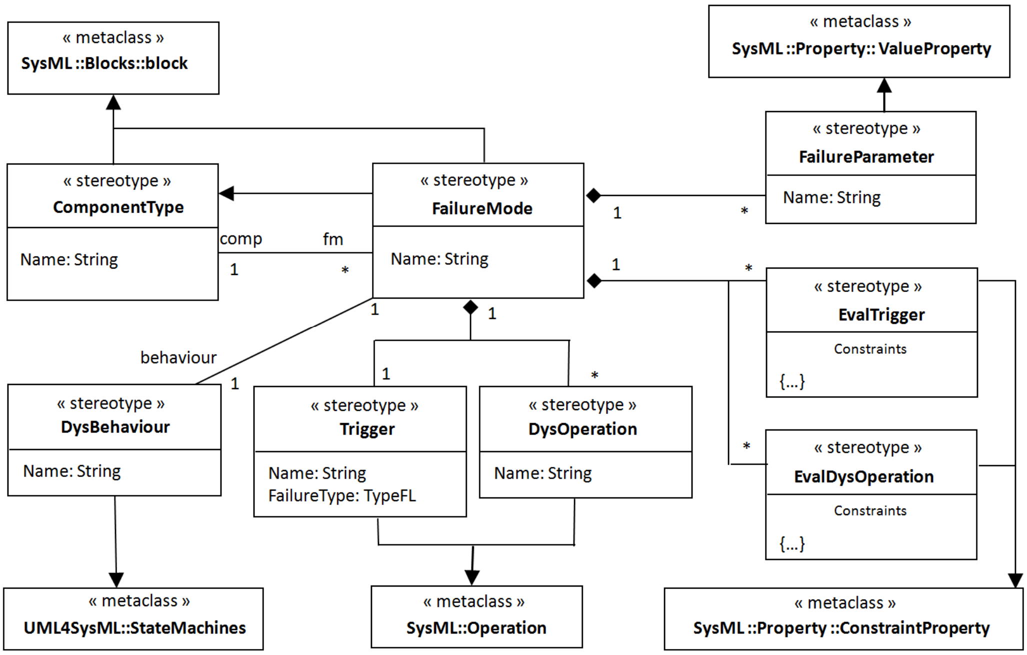


Figure 1 The DBD meta-model

The DBD is organized around the component type. We propose to use the “ComponentType” stereotype showing that a given SysML block (blocks are used to model modular units of system description [6]) is a type of entity we register in our DBD. Then, each failure mode of this type of component is registered using a SysML block stereotyped by “FailureMode”. This is presented in Fig. 1 by the association between the “ComponenType” and “FailureMode”. The black arrow between these two blocks show that “FailureMode” inherits “ComponentType” so that all the behavior and attributes declared for a component are reusable by its failure modes. This construction makes it possible to access the component parameters for its failure modes definition. Failure modes are defined by a set of parameters, operations (showing its behavior) and statemachine (synchronizing behaviors). Diverse stereotypes have been set up to define these elements.

A failure mode is composed of 4 parts: entities used to describe the failure triggering, entities to describe the failure development, a statemachine to synchronize the normal and faulty behavior of the component, and entities stereotyped as “FailureParameters” which express every parameter involved in the failure mode either for triggering, developing or synchronizing. We will detail the way triggering is modeled since it is the part that is most compatible with former reliability databases and quickly summarize how failure development and synchronization are modeled.

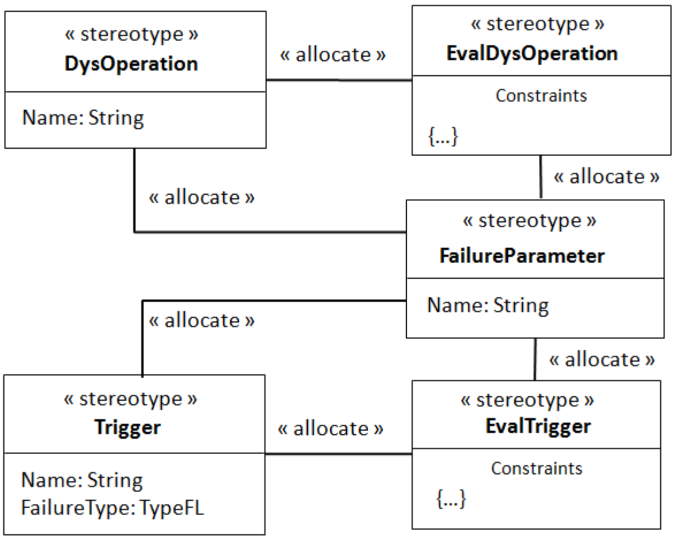


Figure 2 DBD meta-model allocations

The failure trigger is modeled by a SysML operation stereotyped as “Trigger” that possesses a “FailureType” which indicates the type of triggering law followed by the failure mode (i.e. Exponential, Weibull, Lognormal or Systematic). This “Trigger” operation is expressed by its allocated SysML constraint (see Fig. 2; SysML constraints are modeling constructs that express mathematical expressions shared by several parameters) stereotyped by “EvalTrigger” and uses parameters from the various “FailureParameters”. The “FailureType” attribute of the “Trigger” operation permits to define if the failure mode corresponds either to a random failure or to a systematic failure as it is done in the IEC 61508 [3].

In the example of modeling a random failure, the “EvalTrigger” constraint represents the probabilistic law of the trigger (e.g. Exponential, Weibull etc.). Some of the parameters participating to this failure law are specific to the dysfunctional aspect of the component behavior, like the famous λ used in exponential laws or the diverse indicators on component technology and manufacturer used in databases as FIDES. All of them will appear in our DBD as “FailureParameters”. This structure is used to reflect what is available in current reliability databases: Failure probabilities, their associated parameters and the way to compute them; but also to supply the information needed for failure behavior quantification.

Concerning the modeling of dysfunctional behavior description, we assume that when a component becomes faulty, it may execute differently its functions or execute new functions not specified in its mission. These functions are modeled by SysML operations stereotyped by “DysOperation”. As it was done for “Trigger”, these operations are allocated to SysML constraints stereotyped by “EvalDysOperation” to describe the faulty function in detail. The last specific stereotype used in the DBD is made to declare the attached statemachine that synchronizes the normal and faulty behavior of the component. This statemachine is stereotyped “DysBehavior”.

3. THE FIDES GUIDE

3.1. The core methodology

Concerning the reliability databases, we decided to use FIDES [14] as an example since it embeds reliability analysis taking into account various impacting factors and is recognized in the industry [16]. FIDES currently only accounts for electronic components but allow high level of details when describing systems and components. The method we present in this article aims at connecting our DBD and FIDES to embed in our DBD the possibilities of the FIDES guide in terms of reliability analysis.

The FIDES guide 2009 is composed of three parts. The first describes the methodology and the philosophy behind FIDES. The second part contains the actual data for computing components failure rates. The last part details how to control and audit the reliability engineering process. The core methodology of FIDES relies on the principle of calculating the failure rate of the component (λ) for its whole life cycle. This λ will depend on the life phases and various parameters reflecting the impact of the design process, of the manufacturing process, of the manufacturer, etc… This core methodology represents the generic formulas common to every component FIDES deals with. Then, for every family of components, or even for specific components, FIDES extends its core methodology with more precise formulas. Before analyzing the equations that composes the FIDES core methodology, we need to define two terms used within the FIDES guide to define the various granularity levels:

- Product: This refers to the assembled entity for which reliability is being studied.
- Item: In this guide, an item refers to an elementary entity, not broken down, for which the reliability can be studied.
-

Within FIDES, an actual item can be a very specific component such as a resistor or a capacitor.

The main equation of the FIDES methodology is to determine the failure rate of the product through its whole life, λ_{product}:

$$\lambda_{\text{product}} = \sum_{\text{item}} \lambda_{\text{item}} \tag{1}$$

The λ_{item} is composed of 3 factors:

$$\lambda_{\text{item}} = \lambda_{\text{physical}} \times \Pi_{\text{PM}} \times \Pi_{\text{Process}} \tag{2}$$

- $\lambda_{\text{physical}}$ describes to the physical contributions of the failure rate.
- Π_{PM} (PM = Part Manufacturing) represents the item quality. The evaluation method may vary depending on the nature of the item considered.
- Π_{process} represents the quality and technical control over the development, manufacturing and usage process for the product containing the item.

As it is defined by FIDES, Π_{PM} and Π_{process} are multiplication terms that mitigates the purely physicals contributions represented by $\lambda_{\text{physical}}$. The term Π_{PM} symbolizes the quality of the item manufacturing. The term Π_{process} symbolizes the quality and technical control over reliability in the product design process. In complete FIDES analysis, Π_{process} is calculated based on results of a very detailed audit and Π_{PM} is calculated by evaluating the manufacturer's process. FIDES also plans to simply assign a constant value to each of them based on experience in case of a lack of sufficiently detailed data. For example, Π_{process} is by default set to 4 and it is advised to set Π_{PM} to 1.7 for active components and to 1.6 for other components and COTS. We won't describe in details how to determine these multiplication terms because it is part of the component specific part of the FIDES guide [14], and focus on the failure rate.

To determine $\lambda_{\text{Physical}}$ we will combine 2 other formulas given by the FIDES guide:

$$\lambda_{\text{physical}} = \sum_{\text{i}}^{\text{phases}} \left(\frac{\text{Annual\_time}_{\text{phase i}}}{8760} \times \lambda_{\text{phase i}} \right) \tag{3}$$

$$\lambda_{\text{physical}} = \left[\sum_{\text{contributi ons}} \left(\lambda_0 \times \Pi_{\text{acceleration}} \right) \right] \times \Pi_{\text{induced}} \tag{4}$$

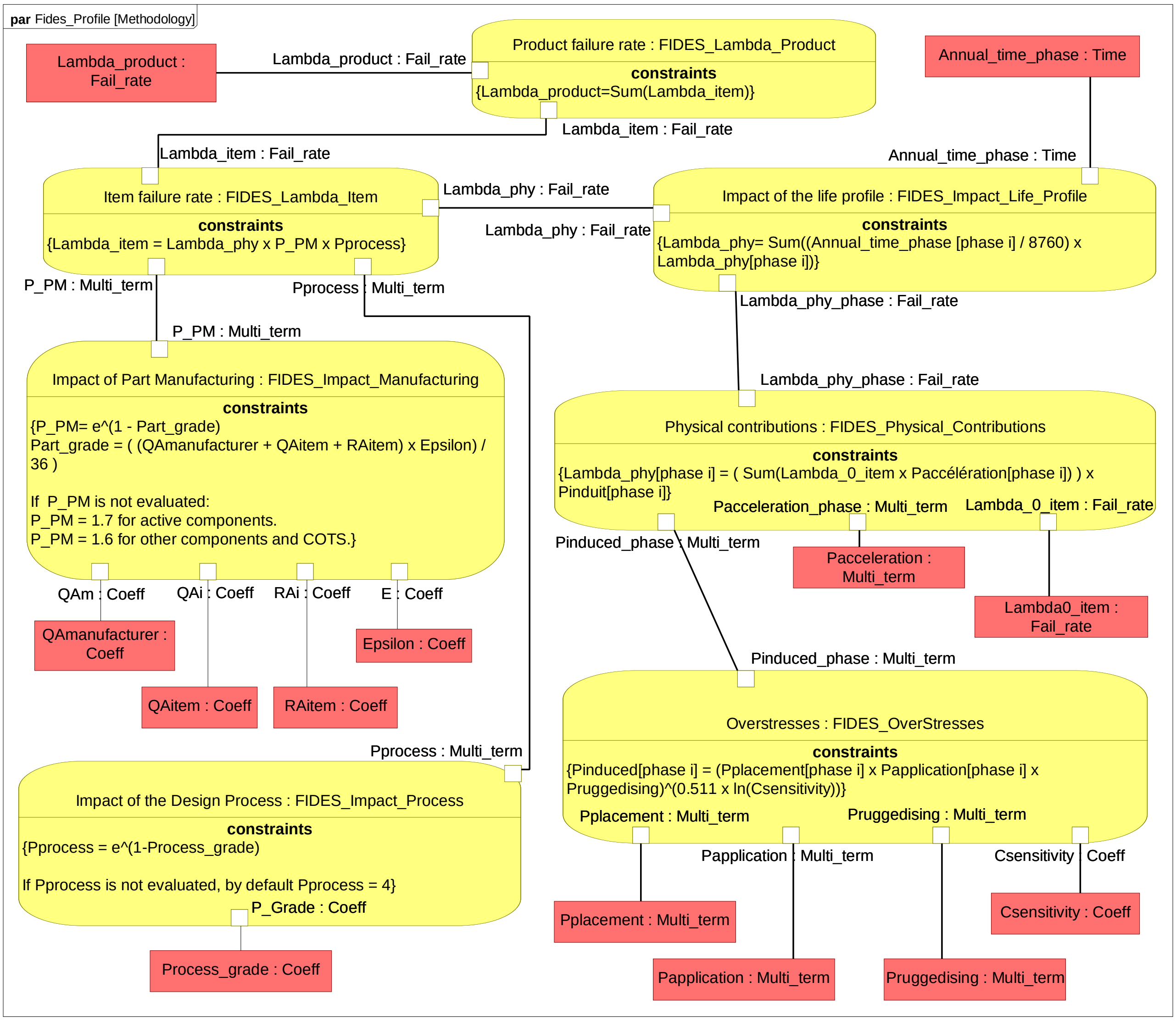
The formula (3) expresses the impact of the life profile on the failure rate for a one year period of time. The parameter Annual_time is the time spent in a specific phase and the $\lambda_{\text{phase i}}$ is the failure rate in the conditions of a specific phase. Formula (4) expresses the impact of various stresses on the basic constant failure rate of the item λ_0. The Π_{induced} translates the sensitivity to usage conditions. It is composed of various factor for each family of physical stresses (thermal, electrical, mechanical, chemical, humidity, temperature cycling). The Π_{induced} represents the overstresses not usually listed as such: item placement in the equipment and usage environment but also parameters such as ruggedizing and sensitivity to overstresses. At first sight we have a contradiction since there is two definition of how to calculate $\lambda_{\text{physical}}$. However FIDES describes Π_{induced} as a factor that depends on the life phases, $\Pi_{accelleration}$ depends on usage conditions and even λ_0 depends on physical stresses. In fact, formula (4) is only relevant in a specific phase. Finally we can say that in formula (4), it is not $\lambda_{physical}$ that is calculated but the $\lambda_{phase\ i}$.

The two most important aspects of the FIDES guide are:

- Both physical and process contributions to risk are calculated,
- The physical contributions are highly dependent on the life cycle definition.

Even without detailing the formulas used to compute $\Pi_{accelleration}$ or $\Pi_{induced}$ that rely on the description of life phases, we can say that adjusting parameters describing the life phases of a product accurately is the reliability expert most important job during FIDES analysis.

To make it easier to analyze this core methodology, we use a SysML model to highlight the connection between these formulas and extract the list of parameters needed to compute the failure rate of a product. In SysML, each formula will be modeled as a constraint property. We will then be able to connect those constraints within a parametric diagram (Fig 3).



par Fides_Profile [Methodology]
Lambda_product : Fail_rate
Lambda_product : Fail_rate
Product failure rate : FIDES_Lambda_Product
constraints
{Lambda_product=Sum(Lambda_item)}
Annual_time_phase : Time
Lambda_item : Fail_rate
Lambda_item : Fail_rate
Annual_time_phase : Time
Item failure rate : FIDES_Lambda_Item
constraints
{Lambda_item = Lambda_phy x P_PM x Pprocess}
Lambda_phy : Fail_rate
Lambda_phy : Fail_rate
Impact of the life profile : FIDES_Impact_Life_Profile
constraints
{Lambda_phy= Sum((Annual_time_phase [phase i] / 8760) x Lambda_phy[phase i])}
P_PM : Multi_term
Pprocess : Multi_term
Lambda_phy_phase : Fail_rate
P_PM : Multi_term
Impact of Part Manufacturing : FIDES_Impact_Manufacturing
constraints
{P_PM= e^(1 - Part_grade)
Part_grade = ((QAmanufacturer + QAitem + RAitem) x Epsilon) / 36)
If P_PM is not evaluated:
P_PM = 1.7 for active components.
P_PM = 1.6 for other components and COTS.}
Lambda_phy_phase : Fail_rate
Physical contributions : FIDES_Physical_Contributions
constraints
{Lambda_phy[phase i] = (Sum(Lambda_0_item x Paccélération[phase i])) x Pinduit[phase i]}
Pacceleration_phase : Multi_term
Lambda_0_item : Fail_rate
Pinduced_phase : Multi_term
Pacceleration : Multi_term
Lambda0_item : Fail_rate
QAm : Coeff
QAi : Coeff
RAi : Coeff
E : Coeff
QAmanufacturer : Coeff
Epsilon : Coeff
QAitem : Coeff
RAitem : Coeff
Pinduced_phase : Multi_term
Overstresses : FIDES_OverStresses
constraints
{Pinduced[phase i] = (Pplacement[phase i] x Papplication[phase i] x Pruggedising)^(0.511 x ln(Csensitivity))}
Pplacement : Multi_term
Pruggedising : Multi_term
Pprocess : Multi_term
Impact of the Design Process : FIDES_Impact_Process
constraints
{Pprocess = e^(1-Process_grade)
If Pprocess is not evaluated, by default Pprocess = 4}
P_Grade : Coeff
Papplication : Multi_term
Csensitivity : Coeff
Pplacement : Multi_term
Csensitivity : Coeff
Process_grade : Coeff
Papplication : Multi_term
Pruggedising : Multi_term

Figure 3 Parametric diagram of the FIDES methodology

To ease the understanding of Figure 3, we will summarize a few rules of SysML parametric diagram. There are 3 types of entities:

- Constraints properties, materialized by round cornered rectangle. It is an instance of a Constraint Block. Constraint block defines a generic formula and the constraint property defines its use in an actual system/context and connected eventually with other constraint properties. For example, “Ohm

Law" would be a constraint block, the matching constraint property in a system could embed the specific formula { U1 = (R1 + R2) × I1 }.

- Constraint parameters, shown as squares on the edge of constraint properties. These represent the parameters involved in the formula of the constraint property. Following with the previous example, U1, R1, R2, and I1 are constraint parameters.
- Value properties, presented by rectangles. These are the properties of components of the system. They are used to show how components properties are linked with each other through constraint properties. For example, U1 could be linked to the property "voltage generated" of a voltage generator.

On this Figure 3, we identify the formulas 1 to 4 as the constraints properties: "Product Failure rate", "Item Failure rate", "Impact of life profile" and "Physical contributions". The other constraints properties represent how to calculate Π_{PM}, $\Pi_{process}$, and $\Pi_{induced}$.

Π_{PM} is calculated from four criteria (where higher number means better results in terms of reliability):

- $QA_{manufacturer}$ is the manufacturer's quality assurance criteria. It is a note between 0 and 3 based on the qualification of the manufacturer.
- QA_{item} is the item quality assurance criteria. It is a note between 0 and 3 based on the certification of the item itself.
- RA_{item} is the item reliability assurance. The FIDES guide defines RA_{item} as relevant only for integrated circuits, discrete semiconductor, LED and optocouplers. It is a note between 0 and 3 based on the tests the item took.
- Epsilon is the experience factor. It represents the component purchaser's experience and trust with the manufacturer. It is in general specific to a manufacturer for all its components but it can be component specific if relevant. Epsilon is a note between 1 and 4.

$\Pi_{process}$ is calculated from the Process Grade. This process grade is obtained through the FIDES audit we mentioned earlier.

The $\lambda_{physical}$ of a phase is calculated with:

- $\lambda_{0\ item}$ which is the intrinsic failure rate of the item relatively to a type of physical stress (thermal, humidity, mechanical,…). It is a value that only depends on the type of the studied item.
- $\Pi_{acceleration}$ which represents the sensitivity of an item to a type of physical stress. It is item specific but is often computed using more detailed parameters of the item and the life phase considered).
- $\Pi_{induced}$ which model the contributions of overstresses on the failure rate.

The factor $\Pi_{induced}$ itself is computed using:

- $C_{sensitivity}$ which represents the coefficient of sensitivity to overstresses inherent to the item technology considered. This value only depends on the type of the item.
- $\Pi_{placement}$ which expresses the influence of the position of the item in the product studied (particularly whether or not it is interfaced).
- $\Pi_{application}$ which represents the influence of the usage of an item. This criterion is evaluated by answering 8 questions concerning the usage in the considered phase.
- $\Pi_{ruggedising}$ which is a factor describing the policy of taking account of overstresses in the product development. This criterion is evaluated by answering questions on the ruggedizing policy.

As a result, Fig 3 permits to identify the properties of components (value properties) that are needed to compute the failure rate of a product with the FIDES methodology. The next step to assure the compatibility between our DBD and FIDES is to link the value properties identified in Fig 3 with the concept of our DBD described in Fig 1.

4. CONNECTING FIDES AND THE DBD

Through our description of the FIDES guide, we presented how to calculate failure rate. This failure rate associated to an exponential failure law corresponds to the operation "Trigger" in our DBD that represents the triggering of a failure. Furthermore, the "EvalTrigger" is defined in the DBD as the constraint property that allows to compute the "Trigger". The "EvalTrigger" constraint property must connect all "Failure Parameters" to handle the calculation of the failure rate λ. In Fig 3, we present the FIDES methodology as a constraint property detailed with a parametric diagram. This parametric diagram connects various value properties with each other to finally compute the failure rate. Ideally, every parameter needed by the FIDES methodology should be stored in the DBD with "Failure Parameters". However Failure Parameters as described within the DBD should be specific to a single type of component but value properties of the FIDES methodology sometimes depends on product design or life profile. As a result, not every needed piece of information can be stored in our DBD. Besides, the core methodology is represented on Fig. 3 but it may vary since several parameters (such as $\Pi_{acceleration}$) may need refinement through additional parametric diagrams depending on the type of component considered. Nevertheless, this doesn't constitute an issue because it only means that EvalTriggers from different component type may differ which is allowed by the DBD definition. Table 3 resumes the conceivable reconnection of the FIDES methodology and our DBD meta-model:

Table 1 Correspondence between FIDES and DBD entities

FIDES entities	DBD entities
Constraint blocks and properties	Used within EvalTrigger
Product Failure rate ($\lambda_{product}$)	Used within EvalTrigger Parameter of Trigger
Life Cycle (Annual Time phase)	NULL
Lambda_0_item(stress) ($\lambda_{0\ item}$)	FailureParameter
Pacceleration(stress) ($\Pi_{acceleration}$)	FailureParameter$^{(1)}$
Csensitivity ($C_{sensitivity}$)	FailureParameter
Pplacement (product design) ($\Pi_{placement}$)	FailureParameter$^{(1)}$
Pruggedising (product design) ($\Pi_{ruggedising}$)	FailureParameter$^{(1)}$
Papplication(phase, product design) ($\Pi_{application}$)	FailureParameter$^{(1)}$
QAmanufacturer ($QA_{manufacturer}$)	FailureParameter
QAitem (QA_{item})	FailureParameter
RAitem (RA_{item})	FailureParameter
Epsilon	FailureParameter
Process_grade	FailureParameter$^{(2)}$

$^{(1)}$: Failure Parameters refined through additional constraint properties depending on component type. Eventually, some of the sub-parameters can be project specific.
$^{(2)}$: Failure Parameters inherited from the company level point of view.
NULL: No correspondent entities.

In Table 3 we can see two types of exceptions that require specific adjustments:

- There are parameters specific to the company used in the FIDES methodology. Parameters such as the process grade should be implemented on the DBD at a company level and instantiate in a project useful version of the DBD. This mechanism of instantiation between company and project levels is not yet formalized. In the meantime the process grade is a Failure Parameter used for every component.
- There are parameters specific to a single project, such as the duration of the phases, the thermal environment in each phases, etc. These parameters should not be stored in the DBD since these data are not reliability parameters. However, they are determined through system engineering when describing the life cycle of the designed system, meaning that these parameters are deductible from a functional model of the system.

Finally, by matching the concept of the DBD and the FIDES reliability database, we point out that system engineering activities lead to defining major aspects of the FIDES analysis (Life profile, Design process, …). We will illustrate that fact in the following example by analyzing how to use both SysML and the DBD to perform FIDES reliability studies.

5. CASE STUDY: RELIABILITY STUDY OF A SIMPLE PRODUCT

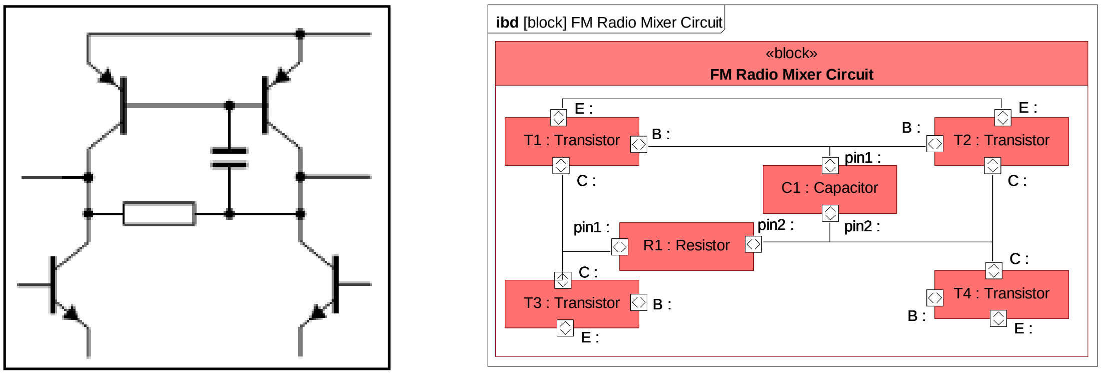


Figure 4 FM Radio Mixer circuit: Electronic schema and SysML internal block diagram

To illustrate the connection of our DBD and the FIDES guide, we realized a FIDES analysis of a simple electric circuit (extract of a radio product Fig. 4) and extracted useful data from the DBD. Our example is an electric circuit from an FM radio. It is composed of 4 transistors, 1 resistor and 1 capacitor as showed on Fig. 4. We suppose that every item is purchased from the same manufacturer and system engineering activities have already been realized resulting in a SysML model (extract on Fig. 4 and Fig. 5). From the item depicted in our SysML internal block diagram on Fig. 4, we are able to search our DBD to get reliability parameters for FIDES analysis.

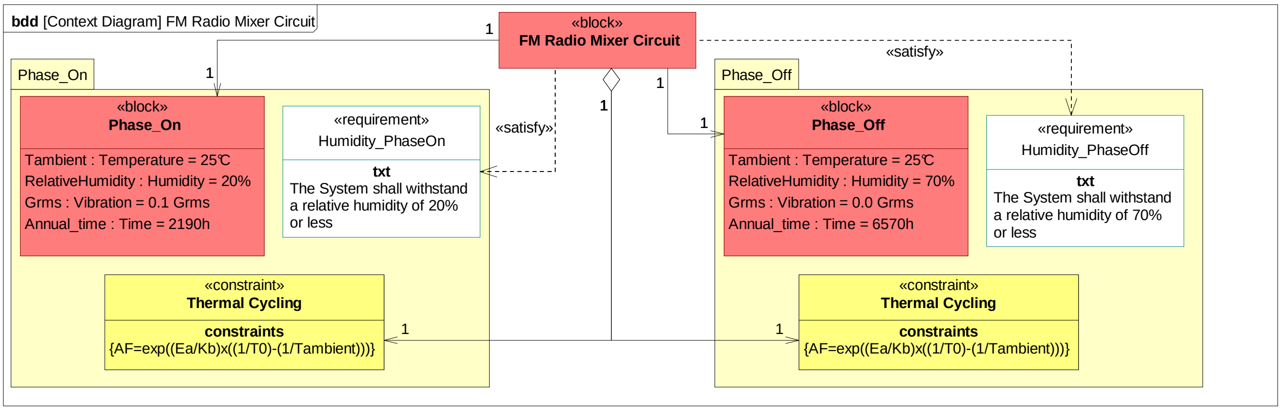

bdd [Context Diagram] FM Radio Mixer Circuit
«block»
FM Radio Mixer Circuit
«satisfy»
«satisfy»
1
Phase_On
«block»
Phase_On
Tambient : Temperature = 25℃
RelativeHumidity : Humidity = 20%
Grms : Vibration = 0.1 Grms
Annual_time : Time = 2190h
«requirement»
Humidity_PhaseOn
txt
The System shall withstand a relative humidity of 20% or less
«constraint»
Thermal Cycling
constraints
{AF=exp((Ea/Kb)x((1/T0)-(1/Tambient)))}
Phase_Off
«block»
Phase_Off
Tambient : Temperature = 25℃
RelativeHumidity : Humidity = 70%
Grms : Vibration = 0.0 Grms
Annual_time : Time = 6570h
«requirement»
Humidity_PhaseOff
txt
The System shall withstand a relative humidity of 70% or less
«constraint»
Thermal Cycling
constraints
{AF=exp((Ea/Kb)x((1/T0)-(1/Tambient)))}

Figure 5 Context Diagram in SysML of the FM radio mixer circuit

Recent works showed how the system functional model of a system can contain life cycle information [6,17]. In our case, a context diagram in SysML permits to describe life phases and their associated requirements, constraints and properties (Fig. 5). The system context and life cycle as modeled during system engineering activities can be instantiated as the life profile needed for FIDES analysis (Table 1). During system engineering activities two phases were considered: On and Off. On Fig. 5, phase properties are described in the blocks Phase_On and Phase_Off. Phase specific constraints and requirement are also present in SysML. The resulting FIDES life profile is summarized on Table 1. We also attributed the mark for $\Pi_{application}$ in each phase.

Table 1 Life profile description

Phase	$t_{annual\text{-}phase}$	Thermal $T_{ambient}$	Humidity RH	Mechanical $G_{RMS\text{-}phase}$	Induced $\Pi_{application}$
Off	6 570h	25 °C	70%	0,0 Grms	2
On	2 190h	25 °C	20%	0,1 Grms	4

Then we can sum up the reliability parameters needed by FIDES that are present in the DBD. The non physical contributions of the risk are composed of the Part Manufacturing and the Design Process. Concerning our design process, we assume we did not had an audit yet and we set $\Pi_{process}$ = 4. The manufacturer of the items is certified ISO 9000 which correspond to $QA_{manufacturer}$ = 1. The items are qualified internally by the manufacturer which correspond to $QA_{component}$ = 1 (The scale for this parameters depends on component type family but internal qualification is equivalent to 1 for resistor, capacitor and transistor). Concerning resistors and capacitors, $RA_{component}$ isn't relevant and set to 0 (if the software used for computing needs a valor). For our transistors, $RA_{component}$ = 1 since it passed designed tests (High Temperature Reverse Bias,…). In our case, the manufacturer is recognized but their processes weren't analyzed which leads to Epsilon = 3. To compute the physical contribution to the risk, two parameters are fully available in the DBD: $C_{sensisivity}$, and $\lambda_{0\ item}$. The $\lambda_{0\ item}$ for each type of stress and the $C_{sensisivity}$ of each item are summarizeded in Table 1.

Table 2 Physical contributions parameters

			transistor	capacitor	resistor
On phase	Thermal	$\lambda_{0\ Th}$	0,014	0,048	0,01
	Humidity	$\lambda_{0\ Rh}$	0	0	0
	Mechanical	$\lambda_{0\ M}$	0,00011	0,0014	0,004
Off phase	Thermal	$\lambda_{0\ Th}$	0	0	0
	Humidity	$\lambda_{0\ Rh}$	0,031	0	0,014
	Mechanical	$\lambda_{0\ M}$	0,00011	0,0014	0,004
	$C_{sensitivity}$		5,2	6,05	3,85

The remaining parameters are deducted from the system engineering activities. The FM radio mixer is not placed as interface in the considered system, resulting in $\Pi_{placement}$ = 1.0 for every item. $\Pi_{ruggedising}$ is set to its default value (1.7) because no study was realized to evaluate it in detail. Finally, $\Pi_{acceleration}$ are computed using both DBD constraint properties and life profile definition parameters. The resulting failure rate calculated is 8.9 FIT (FIT: Failure In Time, 1 FIT = 1 failure per 10^9 hours).

6. CONCLUSION

In this paper we described both our proposition for a model based dysfunctional database and the methodology behind the FIDES guide which is designed as a reliability database. We showed how it is possible to fully benefit from feedback formalized in FiDES within our DBD by connecting the two. The FIDES guide is still a work in progress since new type of components are periodically added to it, making the reach of the FIDES methodology greater. By matching the concept of the DBD and the FIDES reliability database, we also point out that system engineering activities lead to defining major aspects of the FIDES analysis (Life profile, Design process, …). All these make FIDES a worthy target candidate for a process that would extract data from a SysML functional model and use data from the DBD. Such a process would feed one of the software implementing FIDES analysis (Windchill Prediction, RAM Commander, Reliability, Care,… [18]). This process would add up to the MéDISIS framework that aims to ease the interconnection of system engineering and dependability domains.

Currently, MéDISIS and the DBD are applied in the LEA project [4,19], from the project activities planning to the product design phase. The LEA project consists in testing a scram jet motor in real flight conditions which implies a high level of reliability. To address reliability concerns, FIDES analysis will certainly be conducted for critical subsystems of the LEA vehicle. Furthermore, this case study permits to quantify the benefits brought by the use a model-based approach combined with MéDISIS and the DBD.

One of the remaining issues that we need to address is the definition of the different levels of our DBD. As we saw, parameters such as the process grade are not specific to a component but to the entire company. In fact, the information contained in the DBD, as we are foreseeing it, belongs to at least two different levels: the Company DBD (common to every project) and the Project DBD (completed through the project progresses). The process to handle the storage of information, from one level to another, needs to be defined in details and the benefits of each level should be identified.

Finally, addressing the relation between FIDES and our DBD leads us to address the challenges that represent the use of Components Off The Shelf (COTS) in a complex system with high dependability requirements [20]. FIDES currently possess methods to analyze COTS impact on system dependability. These mechanisms will be studied in details to assure that the DBD is fitted to make good use of them.

References

[1] David P & Shawky M. Supporting ISO 26262 with SysML, Benefits and Limits. Proceedings of ESREL 2010, Rhodes, Greece, 2010.

[2] ISO 26262. International Organization for Standardization. Road Vehicles functional Safety. Standard under development.

[3] IEC 61508. International Electrotechnical Commission. Functional Safety of Electrical /Electronic /Programmable Electronic Safety-Related Systems. Parts 1 to 7. 1998-2005.

[4] Cressent R, David P, Idasiak V & Kratz F. Dependability analysis activities merged with system engineering, a real case study feedback. ESREL 2011, Troyes, France, 18-22 September 2011.

[5] Friedenthal S, Moore A, Steiner R. A Practical Guide to SysML : The Systems Modeling Language. The MK/OMG press, Elsevier.2008

[6] Object Management Group, 2010. Systems Modeling Language V1.2, June 2010.

[7] David P, Idasiak V & Kratz F. Reliability study of complex physical systems using SysML. Journal of Reliability Engineering and System Safety, Volume 95, Issue 4, Pages 431-450, April 2010.

[8] Cressent R, David P, Idasiak V & Kratz F. Increasing Reliability of Embedded Systems in a SysML Centered MBSE Process: Application to the LEA Project. 1st M-BED workshop, during DATE 2010, Dresden, Germany, 12 March 2010.

[9] Cressent R, Idasiak V & Kratz F. Mastering safety and reliability in a Model Based process. Proceedings of the 57th Annual Reliability and Maintainability Symposium, RAMS2011, Orlando, Florida, USA, 24-27 January 2011.

[10] Procedure and Data for Estimating Reliability and Maintainability. Report No. M-M-P-59-21, Martin Co., Denver, 1959.

[11] Fragola JR. Reliability and risk analysis data base development: an historical perspective. Reliability Engineering and System Safety, Vol. 51, pp. 125-136, 1996.

[12] Lisagor O, McDermid JA, Pumfrey DJ. Towards a practicable process for automated safety assessment. 26th International System Safety Conference (ISSC), Vancouver, 2008.

[13] Lisagor O, Bozzano M, Breitschneider M, Kelly TP. Incremental Safety Assessment: Enabling the Comparison of Safety Analysis Results. 28th International System Safety Conference (ISSC), Minneapolis, 2010.

[14] FIDES Group. Reliability Methodology for electronic systems – FIDES Guide. May 2009.

[15] USA Department of Defense. Reliability Prediction of Electronic Equipment. Military Standard, MIL HDBK 217f. 1991.

[16] Charpenel P, Davenel F, Digout R, Giraudeau M, Glade M, Guerveno JP, Guillet N, Lauriac A, Male S, Manteigas D, Meister R, Moreau E, Perie D, Relmy-Madinska F, Retailleau P. The right way to assess electronic system reliability: FIDES. Microelectronics Reliability, Volume 43, Issues 9–11, Pages 1401–1404, September–November 2003.

[17] Bassi L, Secchi C, Bonfe M, Fantuzzi C. A SysML-Based Methodology for Manufacturing Machinery Modeling and Design. IEEE/ASME Transactions on Mechatronics, Volume 16, Issue 6, Pages 1049-1062, December 2011.

[18] Windchill Prediction (http://www.ptc.com/product/windchill/quality/),
Reliability (http://www.aldservice.com/en/reliability-products/reliability-software.html),
RAM Commander (http://www.ingenieurwerkstatt.de),
Care (http://www.bqr.com/content/view/15/28).

[19] Falempin F & Serre L. French Flight Testing Program LEA Status in 2009. 16th AIAA/DLR/DGLR International Space Planes and Hypersonic Systems and Technologies Conference, Bremen, Germany, 19-22 October 2009.

[20] RedMill F. Analysis of the COTS debate. Safety Science, Volume 42, Pages 355-367, 2004.