



HAL
open science

Conflict Management for Mechatronic Systems Design

Mouna Fradi, Faïda Mhenni, Raoudha Gaha, Abdelfattah Mlika, Jean-Yves Choley

► **To cite this version:**

Mouna Fradi, Faïda Mhenni, Raoudha Gaha, Abdelfattah Mlika, Jean-Yves Choley. Conflict Management for Mechatronic Systems Design. 2021 IEEE International Symposium on Systems Engineering (ISSE), Sep 2021, Vienna, Austria. pp.1-8, <10.1109/ISSE51541.2021.9582478>. <hal-03836769>

HAL Id: hal-03836769

<https://hal.science/hal-03836769v1>

Submitted on 6 Jun 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Conflict Management for Mechatronic Systems Design

Mouna Fradi

*LMS Laboratory and Quartz Laboratory
University of Sousse and ISAE-Supmecca
Saint-Ouen, France
mouna.fradi@isae-supmecca.fr*

Faïda Mhenni

*Quartz Laboratory
ISAE-Supmecca
Saint-Ouen, France
faida.mhenni@isae-supmecca.fr*

Raoudha Gaha

*LMS and Laboratory of Roberval of Mechanics
University of Sousse and UTC
Compiègne, France
raoudha.gaha@utc.fr*

Abdelfattah Mlika

*LMS laboratory
University of Sousse
Sousse, Tunisie
abdelfattah.mlika@eniso.u-sousse.tn*

Jean-Yves Choley

*Quartz Laboratory
ISAE-Supmecca
Saint-Ouen, France
jean-yves.choley@isae-supmecca.fr*

Abstract—Due to the multidisciplinary and complex nature of mechatronic systems, collaborative design becomes necessary in order to reduce the design cycle time and cost. This collaboration involves the use of numerous languages and expert models in order to represent distinct views on the same system. Consequently, conflicts among these models are likely to happen and need to be detected and resolved carefully. In order to capture and handle these conflicts in an easy and simple way, a unifying formalism based on mathematical concepts is required. Thus, we suggest the use of Category Theory (CT), motivated by its mathematical and formal foundation to handle conflicts in a formal way. This work presents a new framework able to check and manage conflicts in collaborative design. This framework has two key features. First, to detect the conflicts between the expert models in a formal way based on category theory concepts. And second, to manage these conflicts in order to ensure the coherence of the system under design. Our approach is illustrated through its application to an Electronic Throttle Body (ETB) from the automotive industry since it involves the collaboration among several expert domains.

Index Terms—mechatronic design, collaboration, multi-disciplinary design, conflict management, category theory.

I. INTRODUCTION

Mechatronic systems result from the synergic integration of electrical, mechanical, information technology and computer system [1]. This multiplicity of domains renders adopting collaborative processes necessary in order to ensure efficient communication between the different expert models (EMs) relative to each domain [2]. Since these models are performed by several stakeholders, they may present some conflicts that need to be handled carefully [3]. This can be reached through creating a common formalism supporting conflict detection as well as resolution [4]. The formal framework provides, on one hand, the detection of potential contradictions occurred among the expert models, and on the other hand, the resolution of these conflicts.

The choice of the appropriate technique for this issue is critical for attaining the objectives of efficient conflict detection and resolution. In this context, we propose to address

this challenge through the use of a mathematical foundation, that is, category theory (CT). This theory, invented in the early 1940s, provides a structured and organized representation of thoughts using a set of objects related among each other through morphisms [5], [6]. Our main goal remains in handling conflicts between data with fine level of granularity (constraints and parameters) using the formalism of CT.

The present article is organized as follow. Section II presents a review of related works. Section III provides the details of the proposed framework. Then, an application to an Electronic Throttle Body (ETB) is provided in section IV. Finally, section V concludes this paper.

II. LITERATURE SURVEY

A. Conflict management methodologies

The notion of "Consistency management" has appeared in the model-driven software engineering domain by Finkelstein et al. [7]. Taylor et al. [8] suggested that the inconsistency can be described as a contradiction or a conflict between different model elements. The authors proposed five forms of this contradiction: name, interface, behavioral, interaction and refinement inconsistencies. However, only interaction and interface inconsistencies will be studied in the present paper. Interface inconsistency can be defined as mismatching terminologies or values between model elements. This conflict type can be presented when the same parameter has different values or when an "electrical connection", for example, is named "electrical wiring" in another model. However, interaction inconsistency appears when a constraint is not respected. Inconsistencies due to incoherent values or a non-respect of constraints will be considered in our study. Nonetheless, the remaining types of conflicts are out of the scope of the present research work. Several approaches are presented in the related literature. The relevant ones are described below.

Model synchronization approaches have been presented in order to manage conflicts. In this context, Legendre et al.

[9] proposed a model synchronization-based approach. This approach deals with the synchronization of safety analysis architecture design and assessment issues. The synchronization consists in identifying inconsistencies obtained from the confrontation of the different points of view. These conflicts are handled by finding compromises that evolve progressively the different models. An industrial case study is used to validate manually the proposed approach in this work. Consequently, some efforts have been suggested by Berriche et al. [10], [11] in order to automate as well as formalize model synchronization for mechatronic systems. This approach makes use of Query View Transformation (QVT) standard as a support of abstraction and concretization operations. The authors dealt with automated consistency checking. Nonetheless, conflict resolution is still based on a manual process. Model synchronization aims at maintaining consistency among distinct models. However, this approach does not deal with conflict management between fine granularity data, which is the main objective of the present paper.

Another interesting solution has been proposed to deal with conflict management using rules and patterns. Hence, Herzig et al. [12], [13] make use of rules and patterns in order to automate conflict management in the context of Model-Based Systems Engineering (MBSE). The authors proposed to represent heterogeneous models through graphs that can be queried using patterns in order to detect as well as define inconsistencies types. Analogously, Feldmann et al. [14] presented a new approach aiming at handling conflicts. A graphical modeling language is developed in this proposition to detect conflicts between design models. The use of rules in these approaches makes them a flexible solution due to the possibility of modifying these rules without changing the entire conflict management system. Nevertheless, such approaches are based on graphs which can overlook the practice insights and may not keep the right meaning of models used in the collaboration [15].

Additionally, parameters and constraints-based approaches has been proposed in the literature. If, during collaborative design process, a violation of some constraints occurs, the conflict should be, carefully, handled. To do so, Badin et al. [16] presented a new model called *the Knowledge Configuration Model* (KCMModel). It consists of Knowledge Configuration (KC), User Configuration (UC) and Information Core Entity (ICE). In KCMModel, ICEs encapsulates Knowledge, which groups parameters and constraints. The ICE structure helps in detecting conflicts through the interdisciplinary constraints. Several research works present new approaches using KCMModel such as [17], [18]. However, KCMModel showed its limits in conflict handling. Based on the main concepts of KCMModel, Mcharek et al. [19] developed a new model named *Collaborative Design Process and Product Knowledge* (CDPPK). The main objective of CDPPK model is to reorganize parameters in order to ensure efficient collaboration. CDPPK model is inspired by the KCMModel structure proposed in [16]. The authors make use of ICE concept and improved their model by proposing a new entity, that is, *Design Prod-*

uct Knowledge (DPK). DPK contains all the ICEs and the collaboration results. This model suggests resolving conflicts following a manual manner which can be considered as a time-consuming task.

As aforementioned, our main goal consists in formalizing the detection and handling of conflicts occurred between parameters and constraints during collaborative process of mechatronic systems. However, constraint and parameter-based approaches do not provide useful methods to resolve conflicts. Therefore, we propose to integrate the concept of rules and patterns in constraint and parameter methodologies since using rules and patterns will make the process more flexible. However, graphs used in such approaches can overlook the right meaning of the EMs. Thus, formal mathematical foundations become necessary to keep the right meaning and to carry out the intuitions of the practice. CT, the powerful mathematical tool, will be used in this paper to overcome the drawbacks of applying graphs to detect and manage conflicts and to provide a formal framework.

B. Category theory related approaches

Category theory was introduced by Eilenberg and MacLane to explore the relationship among algebra and topological field [5]. A category can be formally defined as a collection of objects and morphisms. Considering a Category \mathcal{C} consisting in [20] (see Fig.1):

- objects (e.g., X, Y, Z).
- A collection of morphisms (e.g., f, g, h) relating the objects among each other.
- An identify morphism $1_X: X \rightarrow X$. This morphism must respect the following property : $f \circ 1_X = f = 1_Y \circ f$.
- A composition morphism $g \circ f: X \rightarrow Z$, where $f: X \rightarrow Y$ and $g: Y \rightarrow Z$.
- An associative property if the composition exists. Considering the morphisms f, g and h , where $f: X \rightarrow Y$, $g: Y \rightarrow Z$ and $h: X \rightarrow Z$, then $h \circ (g \circ f) = (h \circ g) \circ f$.

A functor $F: \mathcal{C} \rightarrow \mathcal{D}$ between two categories \mathcal{C} and \mathcal{D} is a mapping from \mathcal{C} to \mathcal{D} while preserving structure. Here, the functor is considered as a mathematical truth conductor [20].

The powerful mathematical structure of CT is applied in several research works in different domains such as systems engineering and collaborative design. A categorical framework is presented by Zhu and Li [4] in order to design and implement concurrent systems. This approach makes use of functors to check consistency between design categorical models and implementation categorical models. Furthermore, in an extended work presented by Zhu et al. [21], CT was used to formally specify Reactive Autonomous Systems (RAS) and its configuration workflow. The proposed categorical approach can help to check errors in RAS. Additionally, Kibret et al. [22] proposed to formalize the Verifiable Design Process (VDP) using CT. Here, the different models of the VDP are presented through categories. The VDP parts are defined using objects and morphisms. The categorical structures of CT are used to define the different abstraction layers of the VDP.

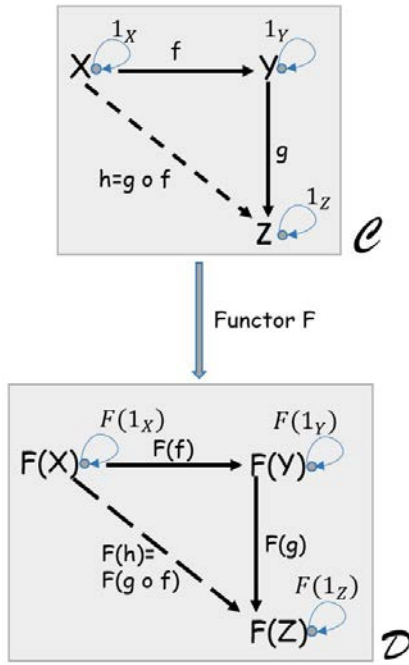


Fig. 1. Category theory basic concepts [20].

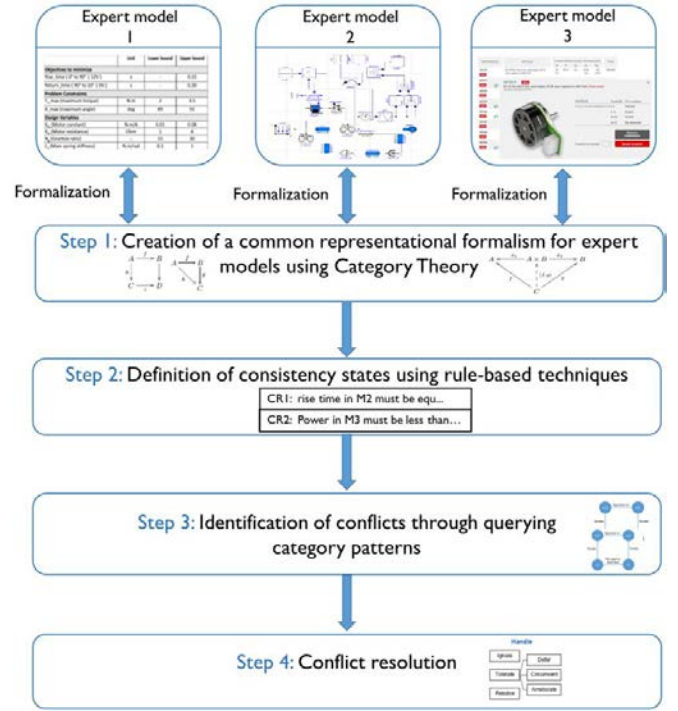


Fig. 2. Proposed conflict resolution process.

In this research paper, CT will provide a formalism that supports conflict checking and management. Compared to the aforementioned works, our proposition represents fine granularity data using the mathematical theory instead of representing the global system. Thus, manipulating mechatronic systems parameters and constrains will facilitate conflict management.

III. PROPOSED APPROACH

Actually, it is common practice for mechatronic collaborative design to make use of a wide range of expert models [23]. Such models are interrelated which may present some conflicts between parameters of these models. The occurred conflicts must be detected and solved at an early stage in order to avoid activity re-execution. Hence, a conflict management methodology is presented in the present study. Our proposition provides a formal conflict checking and resolution in collaborative design. The proposed methodology is based on four main steps, as shown in Fig. 2 : (1) Creating a common representational formalism for expert models based on CT, (2) Defining consistencies using rule-based techniques, (3) Identifying conflicts by querying category patterns and (4) Resolving detected conflicts.

A. Step 1: Creating a CT-based formalism for expert models

Because of expert models heterogeneity, a common formalism becomes necessary. Models are defined as collection of interrelated elements. Thus, graphs provide a formalism generic enough to represent models. However, using graphs in models abstraction may ignore some insights of practice and influence the right meaning of models. Hence, we make

use of CT concepts, motivated by its powerful formalization basis. The mathematical theory provides a common formalism of EMs and carries out practice intuitions. Here, parameters are defined as categories. Each category will contain the evolution of the corresponding parameter from an EM to another. The parameter evolution form an EM to another one, is represented as a set of objects related through morphisms. Our CT-based approach can be introduced as a 7-tuple $F = \{EM_n, C_m, O_p, OL_p, OL_p, M_q, ML_q\}$. EM_n refers to the different expert models (n is the number of the expert models). Parameters used in the different EMs are defined as categories (C_m) (m refers to the total number of parameters). In each category, the instances of the corresponding parameter are considered as objects (O_p) (Here, p is the number of objects in a category). These objects have an identity morphism (OI_p) and labels (OL_p). Parameter instances are related among each other through morphisms (M_q) and have different labels (ML_q) (q is the total number of morphisms in a category). An excerpt of the common formalism based on CT is shown in Fig. 3.

Fig. 3.a illustrates the evolution of a parameter P_1 from the first expert model (EM_1) to the second one (EM_2), whereas, Fig. 3.b refers to the evolution of P_2 from EM_1 to EM_3 . These categories should conform to the formal definition of category mentioned beforehand. Thus, we have to prove the existence of identity, composition and associativity. Let $P1_{EM1}$ and $P1_{EM2}$ two objects representing different instances of the parameter P_1 such that EM_1 interacts with EM_2 through the morphism "activity 1". Let $V1_{EM1}$ and $V2_{EM2}$ two objects to illustrate P_1 values in EM_1 and EM_2 . Each object has an identity morphism, which aims

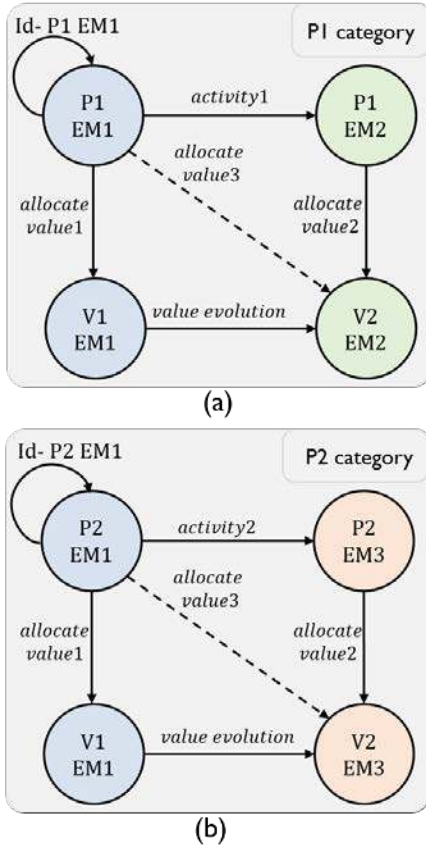


Fig. 3. An excerpt of parameters categories [24].

at representing its internal evolution. We represent this morphism through the looped arrow (see Fig. 3) only for one object to avoid cluttering the representation of categories. $P1_{EM1}$ evolves to $P1_{EM2}$ and $P1_{EM2}$ has a value $V2_{EM2}$. Then, $P1_{EM1}$ can have the value $V2_{EM2}$ indirectly through $P1_{EM2}$ which highlights the morphism of composition. Let “activity 1”, “allocate value 2” and “allocate value 3” be three morphisms where “activity 1”: $P1_{EM1} \rightarrow P1_{EM2}$, “allocate value 2”: $P1_{EM2} \rightarrow V2_{EM2}$ and “allocate value 3”: $P1_{EM1} \rightarrow V2_{EM2}$. It is clear that activity 1 \circ (allocate value 2 \circ allocate value 3) = (activity 1 \circ allocate value 2) \circ allocate value 3.

This verification must be applied to all the created categories in order to ensure the respect of the mathematical definition of a category.

B. Step 2: Defining consistencies using rule-based techniques

The second step of our process remains in defining consistency rules (CRs) in order to describe the constraints to be respected during the design process. A consistency rule can take the following form: $CR_1 = \text{“} P1_{EM2} \text{ must be equal or less than } P1_{EM1}\text{”}$. If one of CRs is not respected, a conflict is detected and must be handled.

C. Step 3: Identifying conflicts by querying category pattern

In this step, the conflicts must be detected. Based on the CRs created previously, the existence of conflicts can be described through a category pattern. These patterns aim at illustrating the conflicts that can exist during the design process. By mapping these patterns categories against each parameter category, conflicts can be captured. This mapping is formally achieved through functors as illustrated in Fig. 4. The morphism between m_1 and m_2 in the category pattern represents the non-respect of a CR.

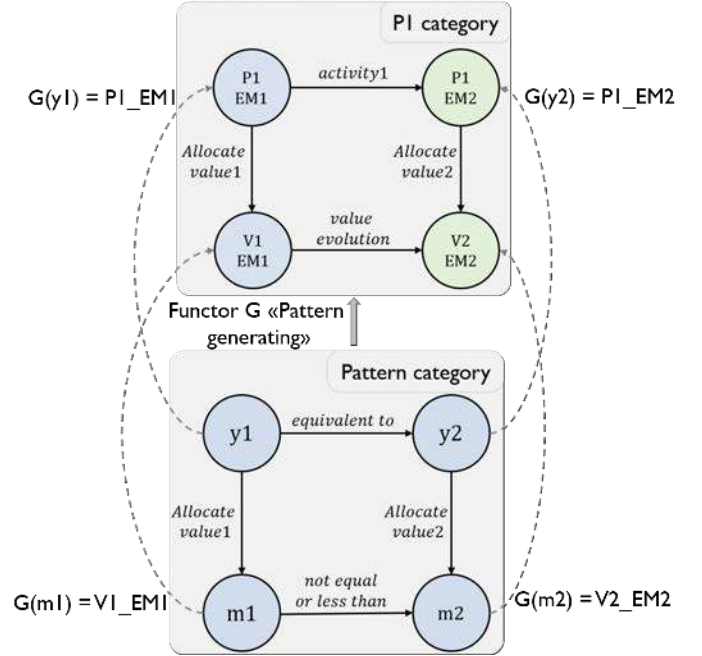


Fig. 4. A mapping between pattern and parameter categories [24].

The image of y_1 , y_2 , m_1 and m_2 through the functor G in pattern category are respectively $P1_{EM1}$, $P1_{EM2}$, $V1_{EM1}$ and $V2_{EM2}$ in P_1 category. By means of this mapping, consistency verification among P_1 values becomes simple and formal. We need, only, to create a functor from pattern category toward the parameter category to detect conflicts.

D. Step 4: Resolving detected conflicts

Once conflicts are detected, an action to handle them must be taken by the project manager. conflict handling can be either tolerating, ignoring or resolving it. The project manager can tolerate a conflict by proposing a slight deviation form the required values or ignore this conflict. However, resolving the conflict means informing the concerned engineers of the detected conflict in order to ameliorate their expert models until finding the appropriate trade-off between them. Let be $P1_{EM1}$ and $P1_{EM2}$, two instances of the rise time defined in requirement model and in the multi-physical model respectively. This conflict can be handled by tolerating it: “ If $P1_{EM2} - P1_{EM1} \leq \alpha$ then tolerate”, where α is the tolerance measure defined by the decision maker.

Thanks to the proposed methodology, it is possible to detect and manage conflicts in the context of mechatronic design. Our approach is considered flexible thanks to the use of CRs, which can be modified or removed without influencing the whole process. In addition, conflict checking is based on formal and mathematical concepts of CT, which helps in carrying out the practice intuitions. To demonstrate the capacity of our approach, an application will be presented in the following section.

IV. CASE STUDY

The Electronic Throttle Body (ETB) is defined as a mechatronic system that regulates the airflow entering to the engine to modify its output torque [19]. ETB system contains a controller, a DC motor, a reducer and a failsafe system composed of two springs, a throttle plate and a position sensor as illustrated in Fig. 5 adopted from [25].

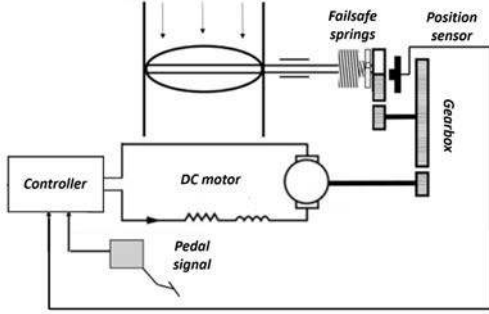


Fig. 5. Electronic Throttle Body architecture adopted from [25].

The motivation behind using the ETB system to validate our methodology is due to its multi-disciplinary nature which helps in illustrating the conflict detection and resolution concept. Thus, our proposed methodology, described beforehand, will be applied to the ETB system.

A. Step 1: Creating a CT-based formalism for expert models

A common formalism for the expert models involved in ETB system design is created in this step. We consider four EMs to illustrate our methodology : (1) a requirement model (EM_R) that identifies the requirements defined for ETB design process, (2) a multi-physical model (EM_{MP}) created to perform dynamic analysis, (3) a Commercial Off-The-Shelf (COTS) components model (EM_C) which contains the characteristics of ETB system components and (4) a 3D model (EM_{3D}) that verifies the mechanism integration. These models will be further described in the following.

(EM_R) is created using SysML language within Magic Draw software [26] in order to specify the requirements to be respected during the ETB design process as described in Fig. 6.

EM_{MP} is a multi-physical model based on Modelica language using Dymola environment [27] as shown in Fig. 7.

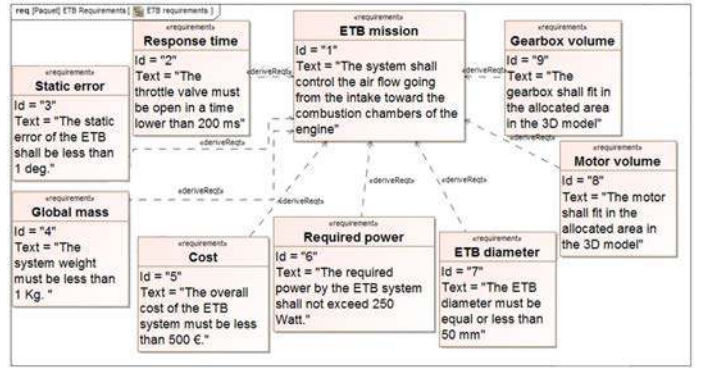


Fig. 6. SysML requirement diagram of the ETB system.

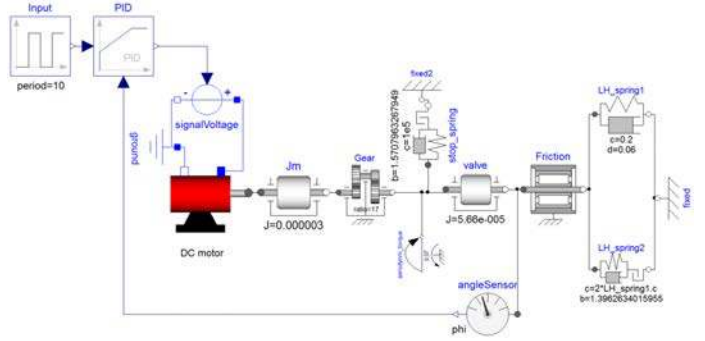


Fig. 7. Multi-physical model of the ETB.

This model contains a DC motor connected to a PID controller, a gearbox which amplifies the torque delivered by the motor, a valve, a friction model, a spring (*Stopspring*) limiting the rotation of the valve between 0° and 90° , an aerodynamic torque representing the pressure difference between the ambient and the combustion pressure and two Limp Home springs. The main spring (*LHspring1*) brings the valve from open position to Limp Home position and the second one brings the valve from closed position to Limp Home position.

Once the multi-physical model is created, the existing Commercial Off-The-Shelf (COTS) components will be chosen conforming within the results of simulation. This model (EM_C) highlights the characteristics of the selected components as show in Table. I.

The last EM is a 3D model within CATIA software (EM_{3D}) [28]. It is established to verify the integration of the selected components in the allocated area (see Fig. 8).

In this paper, a well-defined order to create the EMs must be taken into account. The first phase in the ETB design remains in developing the requirement model (EM_R). Then, the multi-physical model (EM_{MP}) is established. Furthermore, the COTS components in (EM_C) are selected based on EM_{MP} . Finally, the 3D model (EM_{3D}) will provide a verification of selected components integration.

At this level, the required EMs for ETB system design are established. Hence, the common representational formalism is needed. The CT-based formalism illustrates the evolution

TABLE I
COMMERCIAL OFF-THE-SHELF (COTS) MODEL (EM_3) (AN EXCERPT OF DC MOTOR CHARACTERISTICS EXTRACTED FROM [29]).

Values at nominal voltage	
Nominal voltage	12 V
No load speed	8130 rpm
No load current	320 mA
Nominal speed	7610 rpm
Nominal torque	77.7 mN.m
Stall torque	2080 mN.m
Max. efficiency	86%

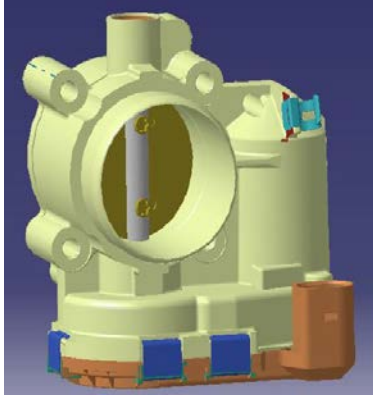


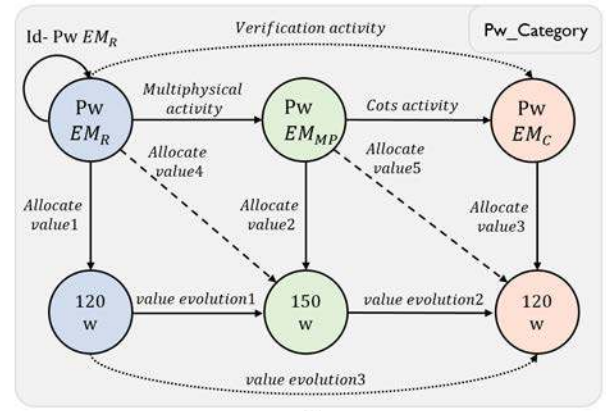
Fig. 8. 3D model of the ETB using Catia environment.

of ETB system parameters. Here, we represent parameters as categories and its different instances as objects. The evolution of this parameter from an EM to another one is established through morphisms. For each parameter a category is created. An excerpt of some parameters categories is illustrated in Fig. 9. The "required power" category is represented in Fig. 9.a. The requirement model (EM_R) provides the first value. This value evolves into new values in EM_{MP} model and the COTS model (EM_C). The identity morphism of each parameter is represented through a looped arrow and highlights its internal evolution. Similarly, Fig. 9.b describes the evolution of the "rise time" parameter (Rt) from the requirement model to the multi-physical model. All the parameters needed for ETB system design are listed in Table II.

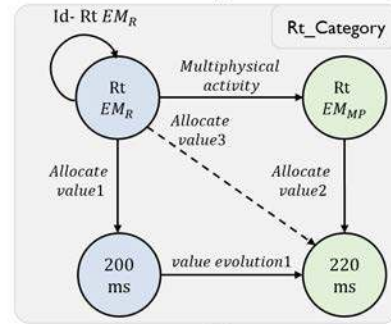
B. Step 2: Defining consistencies using rule-based techniques

The second step in our approach validation consists in defining a set of consistency rules to illustrate the constraints that have to be respected during the ETB design. An excerpt of consistency rules in ETB system is given in the following form:

$$\begin{aligned}
 CR_1 &= Pw_{EM_{MP}} \text{ must be equal or less than } Pw_{EM_R}, \\
 CR_2 &= rg_{EM_C} \text{ must be equal to } rg_{EM_{MP}}, \\
 CR_3 &= Mass_{EM_C} \text{ must be equal or less than } Mass_{EM_R}, \\
 CR_4 &= Rt_{EM_{MP}} \text{ must be equal or less than } Rt_{EM_R}, \\
 CR_5 &= \dots
 \end{aligned}$$



(a)



(b)

Fig. 9. Required power (Pw) and Rise time (Rt) categories in ETB system.

C. Step 3: Identifying conflicts by querying category pattern

Using the CRs defined previously, category patterns have to be created to describe the non-respect of the constraints (see Fig. 4). At this level, all what we need is to match, through functors, the category patterns against each parameter category in order to detect the conflicting values of each parameter. In Table II the conflicting values are shown through the highlighted rows. An example of a mapping among patterns and parameter categories is illustrated in Fig. 10. This mapping will be repeated for all parameters categories.

D. Step 4: Resolving detected conflicts

In this last step, the detected conflicts are managed. As mentioned beforehand, three handling actions (i.e., tolerating, resolving and ignoring actions) can be taken according to the previous experiences and knowledge of the project manager. Table III contains the handling actions taken to manage conflicts appeared during the ETB system design.

The project manager tolerates the conflicts occurred in the "static error" and "global mass" parameters. Moreover, the value mismatch of "cost" and "inertia" parameters is ignored. However, the conflict between the values of the motor power and length parameters are resolved through a close exchange between the involved engineers in the conflict until finding the appropriate trade-off among them.

TABLE II
PARAMETERS USED IN THE ETB DESIGN

Parameters	Symbol	Unit	EM-R	EM-MP	EM-C	EM-3D
Rise time	Rt	ms	200	140	-	-
Static error	Se	deg	0.5	0.6	-	-
Global Mass	Mass	Kg	2.000	-	2.150	-
Max power	Pw	W	250	280	250	-
Cost	Ct	€	1500	-	1555	-
Motor diameter	\varnothing_{mot}	mm	40	-	36	36
Motor Length	Lgth _m	mm	60	-	75	60
Motor resistance	Rm	Ohm	-	4	4	-
Motor Inductance	Lm	mH	-	1.5e-3	1.5e-3	-
Motor Inertia	Jm	Kg.m ²	-	1e-6	1.5e-6	-
Gear diameter	\varnothing_{red}	mm	30	-	30	30
Gear length	Lgth _g	mm	45	-	40	40
Gear ratio	r _g	[]	-	44	44	-

V. CONCLUSION AND FUTURE WORK

A new methodology has been proposed in this research paper in order to manage conflicts during mechatronic systems design. This methodology was based on a combination of fine granularity data and pattern-based approaches. Category Theory was applied in this methodology as a tool for knowledge unification and formalization. CT provided a formal conflict checking using the functor concept in order to map parameter and pattern categories. However, during the conflict management process, parameters dependency was overlooked, which is considered as a critical issue in mechatronic systems. Furthermore, decisions are taken considering the project manager experiences. This makes conflict resolution centralized on its decision.

In future works, an attempt to use CT in parameters dependency representation will be made. The integration of decision-making methods in the presented methodology will be also explored in order to facilitate conflict resolution.

REFERENCES

- [1] F. Mhenni, J.-Y. Choley, F. Caron, and A. Munck, "Collaborative mechatronic design and systems engineering: an educational experiment with karren," in *2018 IEEE International Systems Engineering Symposium (ISSE)*. IEEE, 2018, pp. 1–7.
- [2] F. Mhenni, J.-Y. Choley, and N. Nguyen, "Sysml extensions for safety-critical mechatronic systems design," in *2015 IEEE International Symposium on Systems Engineering (ISSE)*. IEEE, 2015, pp. 242–247.
- [3] I. Dávid, J. Denil, K. Gadeyne, and H. Vangheluwe, "Engineering process transformation to manage (in) consistency," in *COMMITMDE@MoDELS*, 2016, pp. 7–16.
- [4] M. Zhu and J. Li, "Towards a categorical framework for verifying design and implementation of concurrent systems," *Journal of Computer and Communications*, vol. 6, no. 11, pp. 227–246, 2018.
- [5] M. A. Mabrok and M. J. Ryan, "Category theory as a formal mathematical foundation for model-based systems engineering," *Appl. Math. Inf. Sci.*, vol. 11, pp. 43–51, 2017.
- [6] D. I. Spivak, "Category theory for scientists," *arXiv preprint arXiv:1302.6946*, 2013.
- [7] A. C. Finkelstein, D. Gabbay, A. Hunter, J. Kramer, and B. Nuseibeh, "Inconsistency handling in multiperspective specifications," *IEEE Transactions on Software Engineering*, vol. 20, no. 8, pp. 569–578, 1994.
- [8] N. Medvidovic and R. N. Taylor, "Software architecture: foundations, theory, and practice," in *2010 ACM/IEEE 32nd International Conference on Software Engineering*, vol. 2. IEEE, 2010, pp. 471–472.

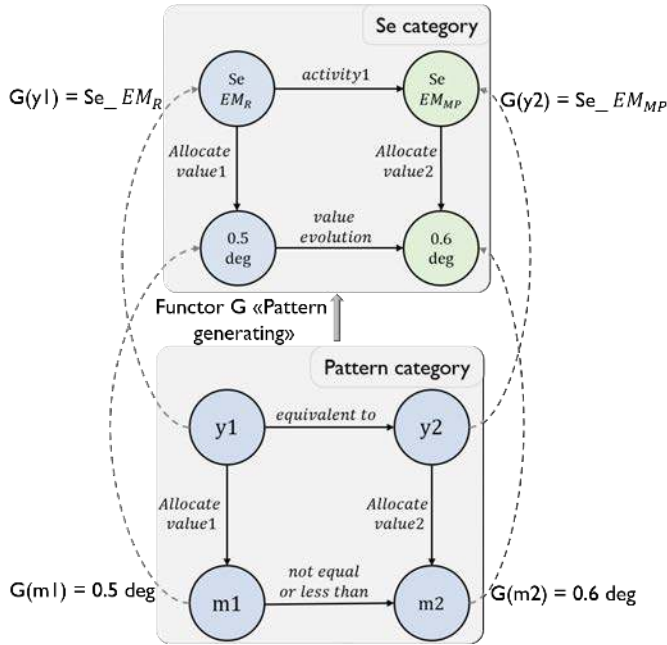


Fig. 10. A mapping between "Static error" parameter (Se) and pattern category.

TABLE III
HANDLING ACTIONS TO MANAGE CONFLICTS

Conflicting values	Handling action
Se_{EMR}, Se_{EMMP}	Tolerate
$Mass_{EMR}, Mass_{EMC}$	Tolerate
Pw_{EMR}, Pw_{EMMP}	Resolve
Ct_{EMR}, Ct_{EMC}	Ignore
$Lgth_{mot_{EMR}}, Lgth_{mot_{EMC}}$	Resolve
Jm_{EMMP}, Jm_{EMC}	Ignore

- [9] A. Legendre, A. Lanusse, and A. Rauzy, "Toward model synchronization between safety analysis and system architecture design in industrial contexts," in *International Symposium on Model-Based Safety and Assessment*. Springer, 2017, pp. 35–49.
- [10] A. Beriche, F. Mhenni, A. Mlika, and J.-Y. Choley, "Towards model synchronization in model driven engineering of mechatronic systems," in *ISSE IEEE*, 2019.
- [11] A. Berriche, F. Mhenni, A. Mlika, and J.-Y. Choley, "Towards model synchronization for consistency management of mechatronic systems," *Applied Sciences*, vol. 10, no. 10, p. 3577, 2020.
- [12] S. J. Herzig, A. Qamar, and C. J. Paredis, "An approach to identifying inconsistencies in model-based systems engineering," *Procedia Computer Science*, vol. 28, pp. 354–362, 2014.
- [13] S. J. Herzig and C. J. Paredis, "A conceptual basis for inconsistency management in model-based systems engineering," *Procedia CIRP*, vol. 21, pp. 52–57, 2014.
- [14] S. Feldmann, K. Kernschmidt, M. Wimmer, and B. Vogel-Heuser, "Managing inter-model inconsistencies in model-based systems engineering: Application in automated production systems engineering," *Journal of Systems and Software*, vol. 153, pp. 105–134, 2019.
- [15] H. Kuang, "Towards a formal reactive autonomic systems framework using category theory," Ph.D. dissertation, Concordia University, 2013.
- [16] J. Badin, D. Chamoret, S. Gomes, and D. Monticolo, "Knowledge configuration management for product design and numerical simulation," in *DS 68-6: Proceedings of the 18th International Conference on Engineering Design (ICED 11), Impacting Society through Engineering Design, Vol. 6: Design Information and Knowledge, Lyngby/Copenhagen, Denmark, 15.-19.08. 2011*, 2011.
- [17] M. Mcharek, T. Azib, M. Hammadi, J.-Y. Choley, and C. Larouci, "Knowledge sharing for mechatronic systems design and optimization," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1365–1370, 2018.
- [18] M. Fradi, R. Gaha, A. Mlika, F. Mhenni, and J. Y. Choley, "Design of an electronic throttle body based on a new knowledge sharing engineering methodology," in *International Conference Design and Modeling of Mechanical Systems*. Springer, 2019, pp. 55–63.
- [19] M. Mcharek, M. Hammadi, T. Azib, C. Larouci, and J.-Y. Choley, "Collaborative design process and product knowledge methodology for mechatronic systems," *Computers in Industry*, vol. 105, pp. 213–228, 2019.
- [20] S. Roman, *An introduction to the language of category theory*. Springer, 2017.
- [21] M. Zhu, H. Kuang, and J. Li, "Representation of categorical specification of self-configurations in reactive autonomic systems framework," *Journal of Computer and Communications*, vol. 6, no. 12, pp. 34–48, 2018.
- [22] N. Kibret, W. Edmonson, and S. Gebreyohannes, "Category theoretic based formalization of the verifiable design process," in *2019 IEEE International Systems Conference (SysCon)*. IEEE, 2019, pp. 1–8.
- [23] R. Ammar, M. Hammadi, J.-Y. Choley, M. Barkallah, J. Louat, and M. Haddar, "Architectural design of complex systems using set-based concurrent engineering," in *2017 IEEE International Systems Engineering Symposium (ISSE)*. IEEE, 2017, pp. 1–7.
- [24] M. Fradi, F. Mhenni, R. Gaha, A. Mlika, and J.-Y. Choley, "Conflict resolution in mechatronic collaborative design using category theory," *Applied Sciences*, vol. 10, no. 11, p. 4486, 2021.
- [25] M. Mcharek, T. Azib, M. Hammadi, C. Larouci, and J.-Y. Choley, "Multiphysical design approach for automotive electronic throttle body," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 8, pp. 6752–6761, 2019.
- [26] MagicDraw, "Dassault systèmes," 2020. [Online]. Available: <https://www.nomagic.com/products/magicdraw>
- [27] Dymola, "Dassault systèmes," 2020. [Online]. Available: <https://www.3ds.com/fr/produits-et-services/catia/produits/dymola/>
- [28] Catia, "Dassault systèmes," 2020. [Online]. Available: <https://www.3ds.com/fr/produits-et-services/catia>
- [29] Maxon, "Maxon group," 2020, last accessed 20 October 2020. [Online]. Available: <https://www.maxongroup.fr/maxon/view/content/index.html>