ASME Open Journal of Engineering
Online journal at:
https://asmedigitalcollection.asme.org/openengineering

# Development and Implementation of a Collaborative Vehicle Configuration Schema in Automotive Engineering

**Charbel Mallah**[1]
German Aerospace Center,
Institute of Vehicle Concepts,
Pfaffenwaldring 38-40,
70569 Stuttgart, Germany
e-mail: Charbel.Mallah@dlr.de

**Marko Alder**
German Aerospace Center,
Institute of System Architectures in Aeronautics,
Hein-Sass-Weg 22,
21129 Hamburg, Germany
e-mail: Marko.Alder@dlr.de

*The design process of road vehicles is a complex endeavor that involves the collaboration of multidisciplinary engineering teams. Each of these teams has its own set of knowledge and tools. As a result, communicating and exchanging design data across these multidisciplinary teams pose significant challenges. Providing a common language for the engineering teams contributes positively to the standardization, efficiency, and robustness of the collaboration within the design process. For this purpose, the aim of this article is to introduce a collaborative vehicle configuration schema (CVeCS). CVeCS acts as a central data model providing a hierarchical parameterization of the vehicle's subsystems, the infrastructure, with which it is interacting as well as the analyses that have been conducted. To demonstrate the advantages of such a data model, the conceptual design phase of the plug-in hybrid electric vehicle named interurban vehicle (IUV) is considered. The IUV is a research vehicle designed at the German Aerospace Centre.*
[DOI: 10.1115/1.4070846 ]

*Keywords: automotive engineering, multidisciplinary design process, collaborative design, collaborative vehicle configuration schema, central data model*

## 1 Introduction

**1.1 Problem Setting.** In recent decades, mankind has experienced an exponential technological growth in various areas. As a result, the complexity of products is increasing significantly. Scientists and engineers are now faced with the challenge of channeling this progress in order to develop economically and ecologically sustainable system architectures. Within this context, transport systems, particularly road vehicles, play a crucial role. A road vehicle is a complex system, which consists of several independent subsystems interacting together. The design of such subsystems and their integration in the overall system requires multidisciplinary know how [1,2]. Moreover, what determines the success of the product, i.e., the road vehicle, is not the optimization of a single subsystem, property, or variable, but rather the optimization of the overall system's performance [3]. Therefore, a holistic approach, which takes into consideration the whole vehicle system and its multidisciplinary character is best suited for the design process. The design process starts with the conceptual design phase, progresses to the detailed design, and finally ends in the manufacturing phase. An essential phase in the design process is the early phase of the vehicle design, also known as the concept phase or conceptual design phase. This phase is crucial as it establishes a major part of the development and production costs of the vehicle [4,5]. With the objective of reducing costs and accelerating the design process, *computer-aided engineering* (CAE) tools are being leveraged, especially during the conceptual design phase. One major challenge in the conceptual design phase is the communication of knowledge across the multidisciplinary teams involved. As each of these teams has its own set of knowledge, CAE tools, and data formats (e.g., text-based formats, binary formats), the collaboration among them becomes a tedious task. This challenge is addressed in the following sections.

**1.2 State of the Art.** This section reviews the state of the art on the conceptual design phase of road vehicles and presents techniques that facilitate the communication and collaboration within the design process, more broadly and beyond the scope of road vehicles.

*1.2.1 Conceptual Design Phase of Road Vehicles.* Within the context of integrating and automating the multidisciplinary efforts within the conceptual design phase of road vehicles, the work presented in Ref. [6] makes use of *multidisciplinary design analysis (and optimization)* (MDA(O)) techniques. Here, CAE tools from

[1]Corresponding author.
Manuscript received October 18, 2025; final manuscript received December 11, 2025; published online February 16, 2026. Assoc. Editor: Cosmin E. Dumitrescu.

various design domains, such as structural design, aerodynamics, interior design, and so on, were integrated into a single design process. The exchange of design parameters across the CAE tools was orchestrated using a relational data base (MS access) as well as a *computer-aided design* (CAD) model for geometric parameters. In addition to the conceptual design phase, MDA(O) techniques are often utilized within the context of crash and *noise, vibration and harshness* (NVH) analysis of road vehicles. See, for instance, the work presented in Refs. [7,8]. Additionally, a substantial body of literature exists that focuses less on the multidisciplinary and collaborative character of the conceptual design phase of road vehicles, while placing increased emphasis on the automation of generating vehicle concepts. For instance, the authors in Refs. [9,10] provide a methodology and a software implementation that generate optimized concepts for electric road vehicles. More recent approaches generate digital vehicle concepts while taking into account autonomous driving as presented by the authors in Ref. [11].

*1.2.2 System Modeling.* Within the context of collaboration across engineering teams, *model-based systems engineering* (MBSE) plays a crucial role in enhancing communication throughout the design process. MBSE represents a specialized approach within the *systems engineering* (SE) framework. While classical SE approaches are based on documents, MBSE utilities digital system models to model, analyze, and verify complex systems, their requirements, and behaviors [12]. Moreover, MBSE aims on creating a centralized system model, which describes the system at hand with all its domain-specific subsystems, thus achieving a single source of truth. To this end, a general-purpose modeling language called *systems modeling language* (SysML), standardized by Object Management Group [13], is used. while SysML is based on the *unified modeling language* (UML), its newest version, SysML v2, is based on *kernel modeling language* (KerML) [14] and provides both a textual and graphical notation of the model.

*1.2.3 Data Modeling via XML.* In addition to SysML, which primary focuses on a generic modeling of complex systems, there exist various data modeling approaches. These aim to define the structure and constraints of the data and its relationships. For instance, the *extensible markup language* (XML) *schema definition* (XSD) [15] is used to define hierarchical data structures. More precisely, an XSD serves as a blueprint, specifying the vocabulary to be used and the semantic rules, the data should be following. A concrete realization of an XSD is represented by an XML document. Based on such data modeling approaches, several attempts were made with the aim of improving communication in the conceptual design phase of vehicles across multidisciplinary teams. Here, one can differentiate between domain-specific and domain-nonspecific data models.

**Domain-Specific Data Models:** Each of the teams involved in the conceptual design phase of vehicles has its own set of computational tools that encapsulate domain-specific expertise. For instance, in the domain of aerospace transportation systems, data models such as *concurrent launch vehicle analysis* (CLAVA) [16] and *common parametric aircraft configuration scheme* (CPACS) [17] were developed for the early design phase of space launchers and aircraft, respectively. Both models define a hierarchical parameterization of the vehicle considered and are implemented using an XSD. Due to this hierarchical structure, the various vehicle systems and their corresponding subsystems and components can be parameterized. Such models are central data models, which provide a single source of truth for all vehicle designers and a common language to integrate their tools into a unified design process [18]. Similar approaches were followed in other fields such as HOLISPEC [19] for the early design phase of maritime transport systems.

**Domain-Nonspecific Data Model:** In contrast to the domain-specific data models such as CPACS, CLAVA, and HOLISPEC, the *common mDO*[2] *workflow schema* (CMDOWS) [20] is used to describe design workflows in a general, domain-nonspecific manner. More precisely, CMDOWS provides a standard for structuring design workflows and defining which domain-specific data are exchanged between the tools integrated in the design workflow. CMDOWS also has a hierarchical structure and is implemented using an XSD.

**1.3 Research Gap and Contribution.** As SysML does not explicitly provide a common language for the engineering teams in the design process, it will not be considered in this article. On the other hand, within the context of aerospace transportation systems, XML-based data models have proven advantageous in collaborative settings and have demonstrated ease of implementation compared with other modeling languages [21]. However, to the authors' knowledge, such data models are not used in the conceptual design phase of road vehicles. For this reason, this article aims on answering the following research question:

*How can an* XML*-based data model be implemented in the conceptual design phase of road vehicles as a single source of truth for all the multidisciplinary teams involved?*

**1.4 Outline.** Section 2 is dedicated for the MDA(O) engineering approach within the context of road vehicles. Section 3 provides an introduction to the *collaborative vehicle configuration schema* (CVeCS). Here, the advantages and disadvantages of CVeCS, its implementation, and its impact on the efficiency of the conceptual design phase of road vehicles are discussed. In Sec. 4, the implementation of CVeCS to the design workflow of the interurban vehicle (IUV) is presented. Finally, a brief summary and an outlook are provided in Sec. 5.

## 2 Multidisciplinary Design Analysis for Road Vehicles

The main objective of the conceptual design phase of road vehicles is to generate consistent gravimetric and volumetric vehicle concepts, which meet the defined high level requirements [22]. Due to the multidisciplinary character of the conceptual design phase, finding vehicle concepts that are consistent and ideally optimal requires solving a set of multidisciplinary equations. For this reason, designers can make use of the MDA(O) engineering approach. To build MDA(O) workflows, engineers from various disciplines contribute using their own set of tools and methods. Figure 1 provides an *extended design structure matrix* (XDSM) [23] of such an MDA(O) workflow. The tools of the disciplines involved are represented as rectangels and are placed on the XDSM diagonal. The numbers 1, 2, 3 represented in the rectangles refer to the execution order of the tools. The data connection between these tools is represented by thick gray lines. More precisely, the input parameters of each tool are placed on the vertical line connected to this tool, whereas the output parameters are placed on the horizontal line. As each discipline is affected by the other in MDA(O), the MDA(O) workflow is evaluated multiple times to reach consistency, which is checked by a converger (rounded rectangle in Fig. 1). For instance, the vehicle performance depends on several parameters including the vehicle's total mass, i.e., for the initial execution of the MDA(O) workflow, an initial total mass is required. In turn, computing the mass of some power train components, such as the battery pack in case of an electric vehicle, depends on the vehicle performance. Based on the calculated mass of the battery pack and the other vehicle's components, the total mass of the vehicle is computed. This computed total mass is then used to reevaluate the vehicle performance. In this manner, the MDA(O) workflow is executed until the total mass of the vehicle converges. Within this context, a central data

[2]MDO stands for multidisciplinary design optimization and refers to a set of domain-specific tools, their data exchange, and their connections.

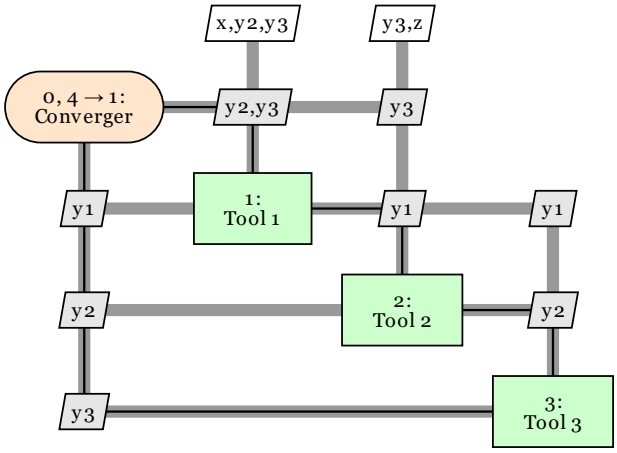



**Fig. 1 XSDM representation of a schematic MDA workflow**

model contributes (i) to the standardization of the interfaces between the domain tools involved and (ii) to enhancing the efficiency of the communication between the domain experts.

## 3 An Introduction to CVeCS

This section introduces the development of CVeCS. CVeCS is a data model inspired by CPACS, which was first introduced in the work presented in Ref. [17]. CPACS has the objective of enhancing the collaboration within the multidisciplinary design process of aircraft. Similarly, CVeCS is used to parametrize road vehicles and orchestrate the data exchange across the disciplines involved in the design phase, thus standardizing the interfaces between them. CVeCS is designed to serve as a central data model that provides vehicle designers with a single source of truth for data in collaborative design activities. Its application reduces the number of interactions and data exchange between the disciplines. To clarify this notion, a maximum number of $N(N-1)$ bidirectional interactions might result, if $N$ disciplines are involved in the design phase. Leveraging a central data model as a single source of truth reduces the number of interactions to $2N$, as illustrated in Fig. 2(*a*) [18]. In the same manner as CPACS, CVeCS is designed as an XSD for use with XML [17]. Therefore, due to the hierarchical structure of XML, CVeCS is able to parametrize the different vehicle systems and their corresponding subsystems and components, as shown in Fig. 2(*a*).

**3.1 Advantages of Extensible Markup Language.** XML is an open-source data modeling language maintained by the World Wide Web Consortium (W3C) and widely recognized and adopted in the field of information technology [18,24]. XML offers multiple benefits including:

- XSD provides a robust validation mechanism. It is used to validate XML documents and check them for consistency, reducing errors and inconsistencies [15].
- XSD is human and machine readable [20].
- The XML format does not depend on a specific programming language and can therefore be integrated with different programming languages [20].
- XML is inherently generic and can thus be used to parameterize road vehicles and their subsystems [18].
- Compared with (system) modeling languages, such as UML, XML offers a simple and extensible approach within collaborative design environments, as mentioned by the authors in Ref. [21].

**3.2 Disadvantages of Extensible Markup Language.** Despite the advantages of XSDs, there exist some disadvantages:

- Large XSDs can become difficult to read and maintain.
- XSDs model system architectures only through data hierarchies, which limits their abilities to decompose system architectures in logical, physical, and functional components.
- XSDs face difficulties in representing complex relationships between elements. For instance, it is challenging to enforce that the value of one parameter depends on or is derived from the value of another.
- XSDs have some limitations when it comes to inheritance and modular design and are not as intuitive as object-oriented models. For instance, multiple inheritance is not supported. This can lead to complex workaround structures.

**3.3 CVeCS Implementation.** As CVeCS is an XSD, the first step of implementing it is to create an XSD file. Such a file defines:

(1) the hierarchical structure of the parameters,
(2) the order in which parameters can be arranged,
(3) the parameters' names,
(4) the parameters' type (e.g., string, integer, array),
(5) the occurrence of the parameters, i.e., how often a parameter may appear.

As a result, a generic vehicle concept is defined. Based on this generic concept, specific vehicle concepts can be derived. Figure 3 represents CVeCS in an XSD diagram, which illustrates the hierarchical structure of the parameters. The occurrences of the elements/parameters can be deduced from the line styles of the element boxes in the XSD diagram. While a dashed line refers to an optional element, see, for instance, `<infrastructure>`, a solid line represents a mandatory element, see, for example, `<header>`. Moreover, if the occurrence of an element/parameter has an upper and lower limit, these limits are shown in the top–right corner of the element box. For instance, $1\ldots\infty$ indicates that the element must appear at least once and may be repeated indefinitely, see, for example, `<engine>`. Additionally, the order of the elements is deduced from the three dots illustrated in the irregular octagon seen in the XSD diagram. If the dots are on top of each other  , the corresponding child elements can appear in any order. Moreover, their occurrence is limited to 0 and 1. On the other hand, if the dots are placed next to each other  , the child elements must appear in a specific order. Here, the child elements can occur from 0 to any number of times [25].

The first level in the hierarchy of the generic vehicle concept consists of five nodes, as shown in Fig. 3. The `<header>` node contains general information mainly regarding the vehicle considered and the version of the XSD being used. The `<vehicle>` node is considered to be the backbone of the data model since it contains most of the vehicle's parameters. Mainly, it contains the following child nodes:

- `<requirements>` node parameterizes high level requirements of the vehicle concept, such as performance and crash requirements.
- `<properties>` node contains some general vehicle properties, such as parameters related to the power train, aerodynamic, inertia, mass, and center of gravity of the entire vehicle.
- `<components>` node parameterizes the main components/subsystems considered for the vehicle concept, such as engines, transmission, exterior, chassis, and batteries. Each of these components consists of its own set of parameters, such as volume, mass, and center of gravity. To support multiple occurrences of the same component (e.g., `<engine>`) while reducing redundancies, each component is uniquely identified using an `id` attribute within the XSD.
- `<energyStorage>` node contains information on how the energy is being stored in the vehicle. This is useful if the considered vehicle is a hybrid one. For instance, 25% of the energy is extracted from the battery pack, while the rest from hydrogen tanks.

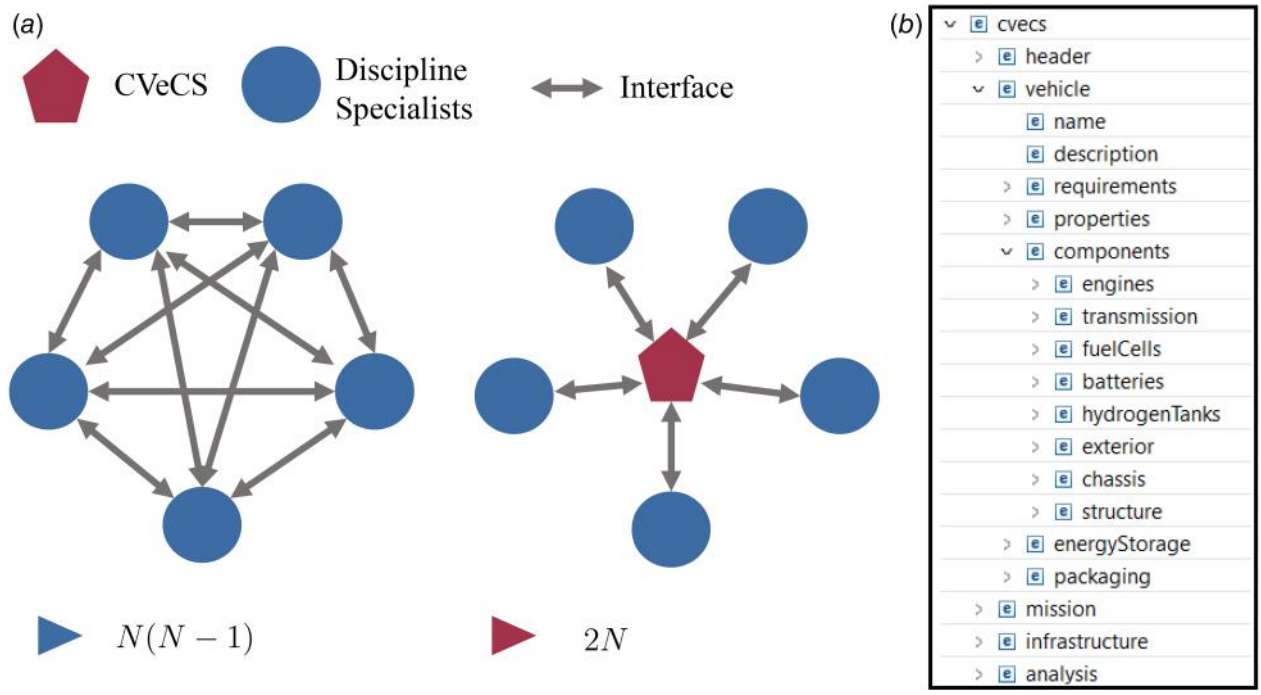



**Fig. 2** (*a*) Illustration of the reduction of bidirectional interactions across engineering domains, achieved by a central data model, such as CVeCS [18]; (*b*) XML realization of CVeCS

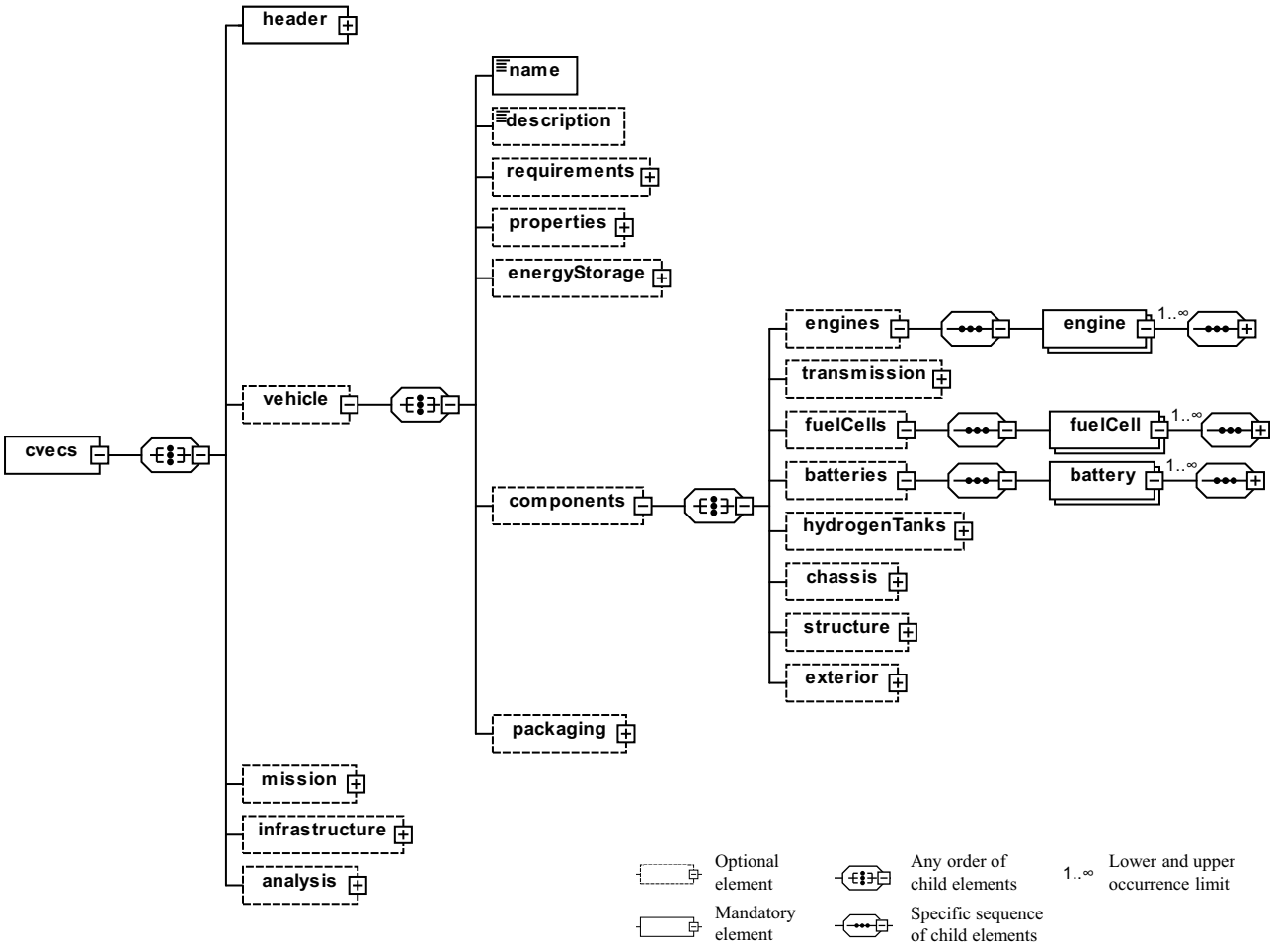



**Fig. 3** XSD diagram of CVeCS

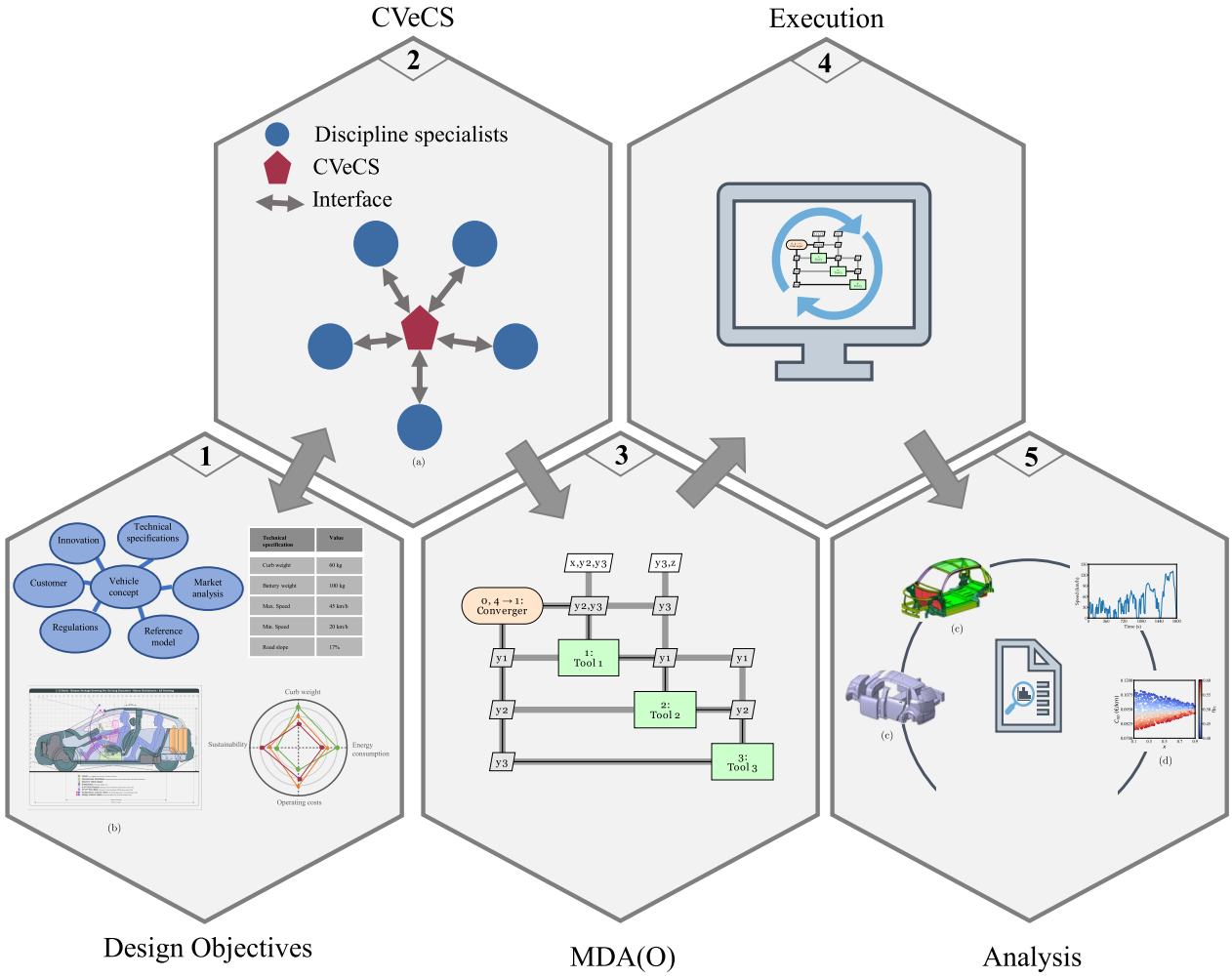



**Fig. 4 The steps involved in creating a CVeCS-based MDA(O) workflow for road vehicles (sources of images (*a*), (*b*), (c), and (*d*): [18,27,26,22], respectively)**

- `<packaging>` node defines how the components are positioned in the vehicle.

As computing the energy consumption of a vehicle concept is essential in the conceptual design phase, a `<mission>` node is added. It consists of parameters related to the driving cycle used for such computation. Additionally, the charging and fueling infrastructure have a large impact on the overall energy balance (well to wheel energy consumption) as well as the operating costs of the considered vehicle. For this reason, an `<infrastructure>` node is added for a parametric description of such data. Finally, an `<analysis>` node is defined to provide a parametric description of the analyses to be conducted by the design workflow (MDA(O)) at hand. In other words, it defines the main outputs to be computed. After creating the XML schema (CVeCS) in an XSD file, an XML file can be derived. For this purpose, there are multiple open-source software (see, for instance, the website[3]), which support creating and/or importing an XSD file and subsequently deriving an XML file from it. Moreover, some of those software provide validation functionalities. More precisely, the parameters' hierarchical structure, order, names, types, and occurrences are validated against the rules defined in the XSD. In order to derive a specific vehicle concept parameterized based on CVeCS, the following steps are required:

(1) Derive an XML file from the XSD file.
(2) In the XML file, add child nodes related to the specific vehicle concept considered. For instance, add a child node `<batteries>` to the parent node `<components>` if an electric vehicle is considered.
(3) Assign values to the elements/parameters in the XML file.

### 3.4 Integrating CVeCS in the Conceptual Design Phase.

Within the context of MDA(O) for road vehicles, CVeCS comes into play with the objective of standardizing the interfaces between the discipline experts involved in the conceptual design phase. The steps involved in creating a CVeCS-based MDA(O) workflow for road vehicles can be described as follows (see also Fig. 4):

(1) **Formulating the Design Objectives:** Here, the parameters relevant for the vehicle concept are defined. This is achieved based on the vehicle's transport task, powertrain configuration (electric, hybrid, etc.), seat configuration, intended operating environment (urban, highway, etc.), required range, and crash scenarios, with which the vehicle must comply [26]. Accordingly, vehicle designers decide which disciplines/domains should be involved and which domain-specific tools are needed. If the existing XSD (CVeCS) covers all the parametric descriptions needed, an XML file

[3]https://eclipseide.org/.

can be directly derived from it, else the XSD should be extended first, before deriving the XML file.

(2) **Developing the Tools' Interfaces to the CVeCS-Based XML File:** Having now a unified language (e.g., names of parameters, unit system) across all teams involved, the interfaces between the tools and the XML file are standardized by developing a wrapper for each tool. Such a wrapper reads the tool's inputs from the XML file and writes the outputs back to the XML file.

(3) **Creating the MDA(O) Workflow:** Here, the tools are fused together into a single MDA(O) workflow. As all parameters are consolidated into a single XMLile, the input and output parameters can be easily assigned to their respective tools (see, for example, *multidisciplinary design analysis and optimization workflow design accelerator* (MDAx) [28]).

(4) **Executing the MDA(O) Workflow:** The execution of the MDA(O) workflow takes place in a process integration software (see, for instance, *remote component environment* (RCE) [29]).

(5) **Processing and Analyzing the Results:** After the execution, the outputs are processed (e.g., visualized) and subsequently analyzed. As the tools' outputs are written to a single XML file, processing the results is facilitated.

**3.5 Impact of CVeCS on the Efficiency of the Conceptual Design Phase.** In order to quantify the impact of CVeCS on the efficiency of the conceptual design phase, the cycle time $\tau_{\text{cycle}}$ is taken into consideration. Within the context of this article, $\tau_{\text{cycle}}$ is defined as the time span between the definition of the requirements and the design objectives to generating a vehicle concept that fulfils these requirements. Additionally, $\tau_{\text{cycle}}$ includes the time needed for rework $R$ due to, for instance, late design changes in the conceptual design phase [30]. Therefore, $\tau_{\text{cycle}}$ can be decomposed into several design tasks as follows:

$$\tau_{\text{cycle}} = \sum_{i=1}^{M} \tau_i + R \tag{1}$$

Here, $M \in \mathbb{N}_{>0}$ denotes the total number of design tasks and $\tau_i$ represents the time needed to conduct the $i$th design task ($i \in \{1, 2, \ldots, M\}$). In this article, the impact of CVeCS on $\tau_{\text{cycle}}$ is examined by comparing the cycle time of the conceptual design phase while using CVeCS ($\tau_{\text{cycle,CVeCS}}$) with the cycle time of the conceptual design phase without using CVeCS ($\tau_{\text{cycle,}\overline{\text{CVeCS}}}$). For this purpose, the following assumptions are made:

- For $\tau_{\text{cycle,CVeCS}}$ and $\tau_{\text{cycle,}\overline{\text{CVeCS}}}$, the five design tasks ($M = 5$) listed in Sec. 3.4 are considered. More precisely, for $\tau_{\text{cycle,}\overline{\text{CVeCS}}}$, the same design tasks are taken into account excluding any tasks and subtasks related to CVeCS and XML files.
- CVeCS as a standard is already developed, i.e., the time needed to develop CVeCS is not included in $\tau_{\text{cycle,CVeCS}}$.
- The domain-specifc tools for the MDA(O) workflow are already developed, i.e., their development time is not included in $\tau_{\text{cycle,CVeCS/}\overline{\text{CVeCS}}}$.

Based on Eq. (1) and the design tasks listed in Section 3.4, $\tau_{\text{cycle,CVeCS}}$ and $\tau_{\text{cycle,}\overline{\text{CVeCS}}}$ can be expressed as follows:

$$\tau_{\text{cycle,CVeCS}} = \tau_{1,\text{CVeCS}} + \tau_{2,\text{CVeCS}} + \tau_{3,\text{CVeCS}} + \tau_{4,\text{CVeCS}} + \tau_{5,\text{CVeCS}} + R_{\text{CVeCS}} \tag{2}$$

$$\tau_{\text{cycle,}\overline{\text{CVeCS}}} = \tau_{1,\overline{\text{CVeCS}}} + \tau_{2,\overline{\text{CVeCS}}} + \tau_{3,\overline{\text{CVeCS}}} + \tau_{4,\overline{\text{CVeCS}}} + \tau_{5,\overline{\text{CVeCS}}} + R_{\overline{\text{CVeCS}}} \tag{3}$$

Here, $\tau_{1,\text{CVeCS}}$ is the time needed to conduct the first design task, i.e., to formulate the design objectives and extend CVeCS, if required (see Sec. 3.4). However, to simplify the comparison between $\tau_{1,\text{CVeCS}}$ and $\tau_{1,\overline{\text{CVeCS}}}$, the time needed for the extension of CVeCS will be included in $R_{\text{CVeCS}}$ and not in $\tau_{1,\text{CVeCS}}$. Therefore, the following equality is valid:

$$\tau_{1,\overline{\text{CVeCS}}} = \tau_{1,\text{CVeCS}} \tag{4}$$

This is due to the fact that, in both cases, the design objectives must be formulated, independent of CVeCS. Moreover, $\tau_{2,\text{CVeCS}}$ in Eq. (2) denotes the time needed to develop the interfaces between the domain-specific tools and the derived XML file based on CVeCS. As a result,

$$\tau_{2,\overline{\text{CVeCS}}} = 0 \tag{5}$$

since no XML file is used. After developing the interfaces, the designers are tasked with connecting the domain-specific tools together to create a unified MDA(O) workflow. The time needed for this task is denoted by $\tau_3$. Without CVeCS ($\tau_{3,\overline{\text{CVeCS}}}$), conducting this task requires an extensive exchange between the domain experts involved in the conceptual design phase. As illustrated in Fig. 2(*a*), there exists a maximum number of $N(N-1)$ interactions between $N$ disciplines/domains. More precisely, the parameters needed for the domain-specific tools must be communicated. Here, the domain experts should ensure consistency, especially in the parameter names, unit system, and coordinate system used. The effort and time required to do so exceed the effort and time needed to create a unified MDA(O) workflow based on CVeCS. When having an XML file, which is derived from CVeCS and contains all the parameters needed for the MDA(O) workflow, the input and output parameters can be easily assigned to their corresponding domain-specific tools. Therefore, fusing the domain-specific tools together into a single MDA(O) is facilitated, for instance, using MDAx [28]. For this reason, the following inequality can be derived:

$$\tau_{3,\overline{\text{CVeCS}}} >> \tau_{3,\text{CVeCS}} \tag{6}$$

Moreover, for simplicity reasons, it is assumed that design task 4, which is executing the MDA(O) workflow, is not influenced by using CVeCS. Therefore, the following equality is valid:

$$\tau_{4,\overline{\text{CVeCS}}} = \tau_{4,\text{CVeCS}} \tag{7}$$

However, one may argue that having a single input and output file for all the domain-specific tools can accelerate the process of reading the inputs and writing the outputs. This fact is partially considered in $\tau_{5,\text{CVeCS}}$. Here, $\tau_5$ refers to the time needed to process and analyze the results. $\tau_5$ can be decomposed into three main subtasks: (i) reading the outputs, (ii) processing (visualization, computation, etc.) the outputs, and (iii) deriving conclusions and decision making. For the comparison between $\tau_{5,\text{CVeCS}}$ and $\tau_{5,\overline{\text{CVeCS}}}$, only the first sub-task is taken into account, since processing, deriving conclusions and decision making are not directly affected by CVeCS. As in the CVeCS case, the inputs and outputs of all domain-specific tools involved are consolidated into a single file, reading them is faster than reading multiple files. For this reason, the following inequality can be deduced:

$$\tau_{5,\overline{\text{CVeCS}}} > \tau_{5,\text{CVeCS}} \tag{8}$$

Finally, for simplicity reasons, the rework time $R$ is defined to be the time needed for late design changes in the conceptual design phase. In the CVeCS case, this could involve the extension of CVeCS to include new parameters. On the other hand, without having a standard as CVeCS, rework requires a larger effort, specifically an extensive exchange with the domain experts. Therefore,

the following inequality is valid:

$$R_{\overline{\text{CVeCS}}} > R_{\text{CVeCS}} \tag{9}$$

Table 1 lists the design tasks and the comparison between their duration with and without CVeCS.

Evaluating the difference between $\tau_{\text{cycle},\overline{\text{CVeCS}}}$ and $\tau_{\text{cycle,CVeCS}}$, i.e., subtracting Eq. (2) from Eq. (3) and taking Eqs. (4), (5), and (7), and inequalities (6), (8), and (9) into account, the following equation is obtained:

$$\begin{aligned}\Delta\tau &= \tau_{\text{cycle},\overline{\text{CVeCS}}} - \tau_{\text{cycle,CVeCS}} \\ &= \underbrace{(\tau_{1,\overline{\text{CVeCS}}} - \tau_{1,\text{CVeCS}})}_{\Delta\tau_1=0} + \underbrace{(\tau_{2,\overline{\text{CVeCS}}} - \tau_{2,\text{CVeCS}})}_{=0} \\ &+ \underbrace{(\tau_{3,\overline{\text{CVeCS}}} - \tau_{3,\text{CVeCS}})}_{=\Delta\tau_3>>0} + \underbrace{(\tau_{4,\overline{\text{CVeCS}}} - \tau_{4,\text{CVeCS}})}_{\Delta\tau_4=0} \\ &+ \underbrace{(\tau_{5,\overline{\text{CVeCS}}} - \tau_{5,\text{CVeCS}})}_{=\Delta\tau_5>0} + \underbrace{(R_{\overline{\text{CVeCS}}} - R_{\text{CVeCS}})}_{=\Delta R>0} \\ &= -\tau_{2,\text{CVeCS}} + \underbrace{\Delta\tau_3}_{>>0} + \underbrace{\Delta\tau_5}_{>0} + \underbrace{\Delta R}_{>0}\end{aligned} \tag{10}$$

Here, $\tau_{2,\text{CVeCS}}$ can be reduced by providing a standard template for the domain experts, on how to establish the interface between domain-specific tools to the XML file. Moreover, based on the experience of CPACS, the time gain by using a central data model in design tasks 3, 5 and the rework time is larger than the time required to establish the interfaces ($\tau_{2,\text{CVeCS}}$). Therefore, the following inequality is satisfied:

$$\underbrace{\Delta\tau_3}_{>>0} + \underbrace{\Delta\tau_5}_{>0} + \underbrace{\Delta R}_{>0} > \tau_{2,\text{CVeCS}} \tag{11}$$

Based on inequality (11) and Eq. (10), one can deduce that:

$$\Delta\tau = \tau_{\text{cycle},\overline{\text{CVeCS}}} - \tau_{\text{cycle,CVeCS}} > 0 \tag{12}$$

Consequently, based on inequality (12), CVeCS decreases the cycle time of the conceptual design phase of road vehicles, thus increasing its efficiency. As this concept has not yet been validated in practice within the automotive engineering sector, the increase in efficiency of the conceptual design process by following a central approach is proven in the aeronautical engineering sector. In the work presented in Ref. [31], a 40% reduction in cycle time was achieved by making use of such a central approach.

## 4 Implementation of CVeCS to the Interurban Vehicle

The IUV is a research vehicle conceptualized and developed within the German Aerospace Center. It was one of the main parts of a project named *Next Generation Car* [32]. The IUV is a five-seater developed with the aim to travel comfortably long distances. It is a plug-in hybrid vehicle equipped with a rechargeable battery pack and a fuel cell. In this article, the applicability of CVeCS will be demonstrated based on the IUV's simplified digital design workflow. This design workflow was used within the context of the IUV's conceptual design phase presented in Ref. [22].

**Table 1 Comparison of the time needed to conduct design tasks in the conceptual design phase with and without the use of CVeCS**

| $i$ | Description | Comparison |
|---|---|---|
| 1 | Formulating the design objectives | $\tau_{1,\overline{\text{CVeCS}}} = \tau_{1,\text{CVeCS}}$ |
| 2 | Developing the tools' interfaces to the CVeCS-based XML file | $\underbrace{\tau_{2,\overline{\text{CVeCS}}}}_{=0} < \tau_{2,\text{CVeCS}}$ |
| 3 | Creating the MDA(O) workflow | $\tau_{3,\overline{\text{CVeCS}}} >> \tau_{3,\text{CVeCS}}$ |
| 4 | Executing the MDA(O) workflow | $\tau_{4,\overline{\text{CVeCS}}} = \tau_{4,\text{CVeCS}}$ |
| 5 | Processing and analyzing the results | $\tau_{5,\overline{\text{CVeCS}}} > \tau_{5,\text{CVeCS}}$ |
| – | Rework | $R_{\overline{\text{CVeCS}}} > R_{\text{CVeCS}}$ |

**Table 2 High level requirements used for the IUV's simplified design workflow [22]**

| Name | Symbol | Value | Unit |
|---|---|---|---|
| Acceleration time from 0 | $t_{0-100}$ | 8.4 | s |
| Top speed | $v$max | 180 | km/h |
| Cruise speed | $v$cont | 160 | km/h |
| Range | $R$ | 1000 | km |

### 4.1 Revisiting the Interurban Vehicle's Simplified Design Workflow.

The main objective of the IUV's simplified design workflow was to digitize and automate its multidisciplinary conceptual design phase. The high level requirements considered within this design workflow are listed in Table 2. Moreover, the simplified design workflow consists of the following engineering domains: vehicle performance, engine sizing, fuel cell sizing, mission, energy storage, mass computation, cost computation, and finally, well-to-wheel energy consumption. As a result, the design workflow consists of 84 design variables, 62 of which are inputs and 22 are outputs.

### 4.2 Interurban Vehicle's Dataset.

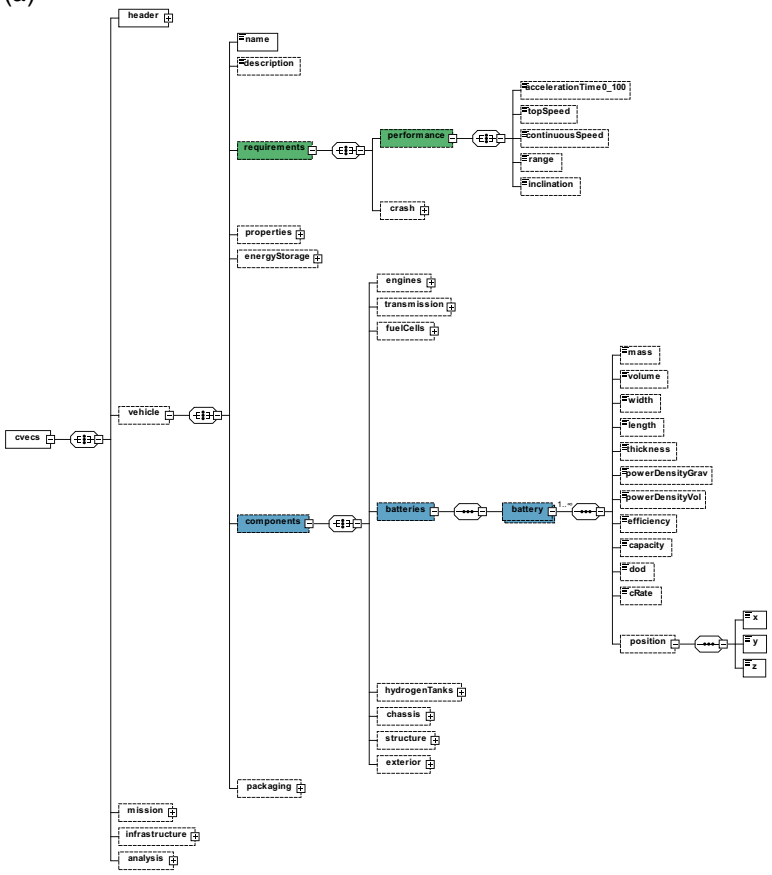
Based on CVeCS, which provides a parametric description of a generic vehicle concept as discussed in Sec. 3.3, the IUV's dataset was derived. More precisely, an XML file was derived where the following elements were added to the XML file:

- IUV's high level requirements with their respective parameter values under `<vehicle>/<requirements>/<performance>` (see Fig. 5).
- The components of the IUV's power train, such as battery and fuel cell were appended with their respective parameter values under `<vehicle>/<components>` (see Figure 5).
- The considered analyses, such as the computation of the energy consumption and operating costs were added under `<analysis>`.

### 4.3 Interurban Vehicle's Multidisciplinary Design Analysis.

The MDA workflow of the IUV was schematically modelled in MDAx (see Fig. 6). MDAx was introduced by Ref. [28] with the aim of simplifying the process of building complex engineering workflows. In MDAx, tools can be created schematically and are represented by rectangles. One of the advantages of MDAx is that it supports the usage of XML-based data models. By uploading the IUV's XML file to MDAx, the input and output parameters can be assigned to the corresponding tools. In this manner, the MDA workflow of the IUV was built. Finally, the MDA workflow was exported from MDAx to the open-source software RCE [29]. RCE is a process integration framework, in which the MDA workflow is automatically executed such that each tool, hosted on its respective server, is invoked, and the necessary data are exchanged [29]. Here, the tools involved extract the inputs needed from the IUV's XML file and write their results back into the XML file. Therefore, the communication between the domain-specific tools involved is standardized.

(*a*)




(*b*)

| Node | Content |
|---|---|
| cvecs | |
| header | |
| vehicle | |
| name | Interurban vehicle |
| description | research vehicle, plug in hybrid |
| requirements | |
| performance | |
| accelerationTime0_100 | 8.4 |
| topSpeed | 180 |
| continuousSpeed | 160 |
| inclination | 0 |
| range | 1000 |
| crash | |
| properties | |
| components | |
| engines | |
| transmission | |
| fuelCells | |
| batteries | |
| battery | |
| id | battery1 |
| mass | |
| volume | |
| width | |
| length | |
| thickness | |
| powerDensityGrav | 290 |
| powerDensityVol | 380 |
| efficiency | 0.80 |
| capacity | |
| dod | 0.93 |
| cRate | 4 |
| position | |
| x | |
| y | |
| z | |
| hydrogenTanks | |
| exterior | |
| chassis | |
| structure | |
| energyStorage | |
| packaging | |
| mission | |
| infrastructure | |
| analysis | |

**Fig. 5** (*a*) XSD diagram of CVeCS providing a parametric description of a generic road vehicle concept (parameterization of high level requirements and battery are highlighted); (*b*) IUV's XML implementation of CVeCS

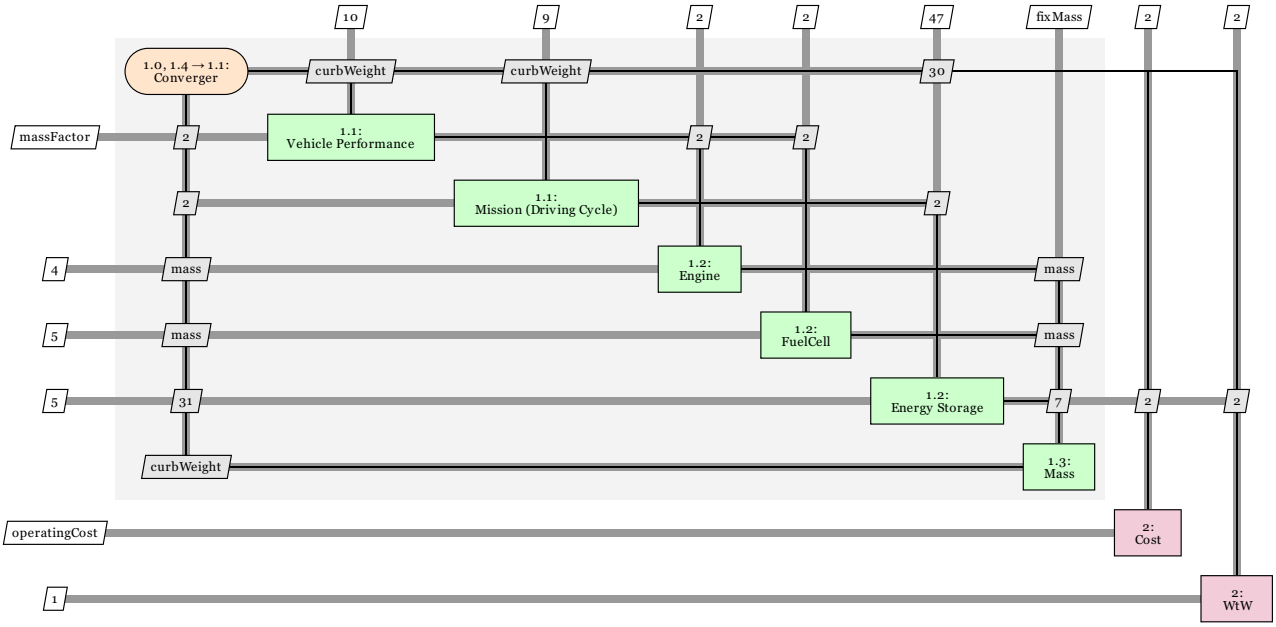



**Fig. 6** XSDM representation of the IUV's MDA workflow

**4.4 Discussion.** Despite its simplicity, the IUV use case provides a proof of concept that demonstrates the added value of a central data model within the context of the conceptual design phase of road vehicles. Some of the added values are summarized and listed below.

- CVeCS serves as single source of truth for all designers involved in the conceptual design phase. It centralizes the access of data needed for the design phase by combining it into a single file. One of the positive implications of having a single source of truth is data consistency. Unit systems and coordinate systems serve as a good example of that. Having access to the same data ensures that all involved domains are using the same unit and coordinate system.
- Moreover, CVeCS enhances the efficiency of the interactions between domain-specific tools. For instance, the tool *Mass* requires input data from the tools *Engine*, *FuelCell,* and *EnergyStorage* (see Fig. 6). Without a central data model, the tool *Mass* is dependent on these other tools and must retrieve its input data by accessing those tools directly or through their output files. By making use of a central data model, *Mass* extracts its input data from the central data model and writes its output back to it. As a result, not only output data of domain-specific tools is consolidated in one file but also the direct dependency of the tool *Mass* on the other tools is eliminated. This not only reduces the number of interactions between the domain-specific tools but also improves their ability to be used as stand-alone tools.
- Additionally, standardizing the interface between the discipline-specific tools and the central data model facilitates the automation of the design process. In other words, as all tool developers use the same interface, integrating the domain-specific tools into a single design process requires less effort.
- Also, combining an MDA(O) approach, which enables simultaneous designing, involving all domains at once, with a central data model, centralizing the communication between those domains, serves as a powerful framework to create a collaborative and modular design process. More precisely, domain-specific tools can be easily extended or replaced in the MDA(O) workflow without affecting other tools. This is due to the elimination of the direct dependency on other tools by the central data model. For instance, the tool *FuelCell* can be easily replaced by another tool, which models an internal combustion engine instead.

## 5 Summary and Outlook

Within the context of this article, CVeCS was developed based on CPACS. CVeCS is an XSD, which provides a hierarchical parameterization of a generic vehicle concept. Such parameters describe the vehicle concept's requirements, components (e.g., engine, chassis), properties (e.g., curb weight, moment of inertia), packaging configuration, energy storage system, mission (driving cycle), as well as the infrastructure with which the vehicle is interacting (e.g., charging infrastructure). Additionally, the analyses that are conducted by the designers are included. Within this context, the parameters' names, types, order, and occurrences are defined by CVeCS. Based on this XSD, a specific vehicle concept represented in an XML format is derived. This XML file serves as a single input and output file for the design workflow. CVeCS functions as a central data model providing a common language for the engineering teams involved in the conceptual design phase of road vehicles.

As a set of multidisciplinary equations must be solved to generate consistent road vehicle concepts, the entire conceptual design phase can be regarded as an MDA(O) workflow. By making use of this central approach, i.e., CVeCS, the interfaces between the multidisciplinary tools within the MDA(O) workflow are standardized. Therefore, the number of interfaces between the domain experts is reduced enhancing the communication efficiency along the conceptual design phase. Moreover, the modularity and flexibility of the design workflow is increased by eliminating interdependencies among the domain-specific tools. To ensure the applicability of CVeCS, the simplified design workflow of the IUV was used (see Sec. 4.3 in Ref. [22]). The IUV's MDA workflow was modeled in MDAx and executed in the open-source software RCE.

One of the most difficult challenges within the context of CVeCS is involving experts from different domains and ensuring that they keep following the same standards. Additionally, finding compromises between those domain experts can become a tedious task since each team of experts has its own priorities and expectations.

In future studies, the increase in efficiency of the conceptual design phase of road vehicles, provided by CVeCS, will be validated by applying it to practical projects. In addition, further development must not be limited to conventional road vehicles, but also take into account nonconventional ones, like, for instance, self-driving cars. Moreover, the applicability of CVeCS within the context of geometric modeling of road vehicles will be explored. Additionally, establishing a GitLab repository under an open-source license to manage version control and enhancements of CVeCS will be considered. This not only promotes the adoption of CVeCS across various projects but also contributes to the enhancements of CVeCS.

## Conflict of Interest

There are no conflicts of interest. This article does not include research in which human participants were involved. Informed consent not applicable. This article does not include any research in which animal participants were involved.

## Data Availability Statement

The datasets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request.

## References


[1] Hirz, M., 2011, "An Approach of Multidisciplinary Collaboration in Conceptual Automotive Development," Inter. J. Collab. Enter., **2**(1), pp. 39–56.
[2] Krog, J., Şahin, T., and Vietor, T., 2022, "Towards a Systems Engineering Methodology for Architecture Development of Vehicle Concepts," DS 118: Proceedings of NordDesign 2022, Copenhagen, Denmark, Aug. 16–18, pp. 1–12.
[3] Braess, H.-H., and Seifert, U., 2013, *Vieweg Handbuch Kraftfahrzeugtechnik*, Springer, Germany.
[4] Schelkle, E., and Elsenhans, H., 2000, *Integration innovativer CAE-Werkzeuge in die PKW-Konzeptentwicklung*, Vol. 14, VDI, Germany, pp. 481–498.
[5] Ehrlenspiel, K., 2007, "Integrierte Produktentwicklung–Denkabläufe Methodeneinsatz Zusammenarbeit," München: Hanser.
[6] Fenyes, P., Donndelinger, J., and Bourassa, J.-F., 2002, "A New System for Multidisciplinary Analysis and Optimization of Vehicle Architectures," 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, GA, Sept. 4–6, p. 5509.
[7] Duddeck, F., 2008, "Multidisciplinary Optimization of Car Bodies," Struct. Multidiscipl. Optim., **35**(4), pp. 375–389.
[8] Kodiyalam, S., Yang, R., Gu, L., and Tho, C.-H., 2004, "Multidisciplinary Design Optimization of a Vehicle System in a Scalable, High Performance Computing Environment," Struct. Multidiscipl. Optim., **26**(3), pp. 256–263.
[9] Kuchenbuch, K., 2012, *Methodik zur Identifikation und zum Entwurf eigenschaftsoptimierter Elektrofahrzeuge*, Logos Verlag Berlin, Germany.
[10] Wiedemann, E., 2014, *Ableitung von Elektrofahrzeugkonzepten aus Eigenschaftszielen*, Cuvillier Verlag, Germany.
[11] König, A. P., 2023, *Methodik zur Auslegung von autonomen Fahrzeugkonzepten*, Technische Universität München, Germany.
[12] Eigner, M., Gilz, T., and Zafirov, R., 2012, "Interdisciplinary Product Development-Model Based Systems Engineering," PLM Portal.
[13] Object Management Group, 2007, "SysML v1 Specification," https://www.omg.org/sysml/sysmlv1/, Accessed August 25, 2025.
[14] Object Management Group, 2025, "About the Kernel Modeling Language Specification Version 1.0 beta," https://www.omg.org/spec/KerML/1.0/Beta1/About-KerML#:%E2%88%BC:text=The%20Kern%el%20Modeling%20Language%20(KerML,PDF, Accessed August 28, 2025.

[15] Fallside, D. C., and Walmsley, P., 2004, "XML Schema Part 0: Primer Second Edition," https://www.w3.org/TR/xmlschema-0/, Accessed March 27, 2025
[16] Fischer, P. M., Deshmukh, M., Koch, A., Mischke, R., Martelo Gomez, A., Schreiber, A., and Gerndt, A., 2018, "Enabling a Conceptual Data Model and Workflow Integration Environment for Concurrent Launch Vehicle Analysis," 69th International Astronautical Congress (IAC), Bremen, Germany, Oct. 1–5.
[17] Nagel, B., Böhnke, D., Gollnick, V., Schmollgruber, P., Rizzi, A., La Rocca, G., and Alonso, J. J., 2012, "Communication in Aircraft Design: Can We Establish a Common Language?" 28th International Congress of the Aeronautical Sciences, Brisbane, Australia, Sept. 23–28.
[18] Alder, M., Moerland, E., Jepsen, J., and Nagel, B., 2020, "Recent Advances in Establishing a Common Language for Aircraft Design With CPACS," Researchgate.
[19] Harries, S., and Abt, C., 2021, "Integration of Tools for Application Case Studies" *A Holistic Approach to Ship Design: Volume 2: Application Case Studies*, Vol. 2, Springer, Switzerland, pp. 7–45.
[20] van Gent, I., La Rocca, G., and Hoogreef, M. F., 2018, "CMDOWS: A Proposed New Standard to Store and Exchange MDO Systems," CEAS Aeronaut. J., **9**(4), pp. 607–627.
[21] Böhnke, D., Litz, M., and Rudolph, S., 2010, "Evaluation of Modeling Languages for Preliminary Airplane Design in Multidisciplinary Design Environments," Congress of the Deutsche Gesellschaft für Luft- und Raumfahrt, Hamburg, Germany, Aug. 31–Sept. 3.
[22] Mallah, C., David, C., and Alder, M., 2024, "Probabilistic Modelling of the Conceptual Design Phase in Automotive Engineering," Proceedings of IAC in Budapest 2024, Budapest, Hungary, Dec. 1, pp. 267–289.
[23] Lambe, A. B., and Martins, J. R., 2012, "Extensions to the Design Structure Matrix for the Description of Multidisciplinary Design, Analysis, and Optimization Processes," Struct. Multidiscipl. Optim., **46**(2), pp. 273–284.
[24] Vonhoegen, H., 2002, *Einstieg in XML*, Vol. . 5, Galileo Press, Germany.
[25] W3Schools, 2025, "XML Schema Reference," https://www.w3schools.com/xml/schema_elements_ref.asp, Accessed December 8, 2025.
[26] Münster, M., 2020, *Vorgehensmodell zur Grundkonzeption eines Fahrzeugkonzepts und Entwicklung neuartiger kraftflussoptimierter Karosseriestrukturen für elektrifizierte Fahrzeuge*, Universität Stuttgart, Germany.
[27] Simulator Hutchinson, 2012, "Human Packaging for the Sim Rig," https://hutchinsonsimulators.wordpress.com/2012/01/11/human-packaging-for-the-sim-rig/, Accessed February 27, 2025.
[28] Page Risueño, A., Bussemaker, J., Ciampa, P. D., and Nagel, B., 2020, "MDAx: Agile Generation of Collaborative MDAO Workflows for Complex Systems," AIAA Aviation Forum, Virtual Event, June 15–19, p. 3133.
[29] Boden, B., Flink, J., Först, N., Mischke, R., Schaffert, K., Weinert, A., Wohlan, A., and Schreiber, A., 2021, "RCE: An Integration Environment for Engineering and Science," SoftwareX, **15**, p. 100759.
[30] Gomes, M., Correia, M., Marques, F., and Alves, A. C., 2023, "Lean Product and Process Development in an Automotive Company," Preprints.
[31] Ciampa, P. D., and Nagel, B., 2020, "AGILE Paradigm: The Next Generation Collaborative MDO for the Development of Aeronautical Systems," Prog. Aerosp. Sci., **119**, p. 100643.
[32] Institute of Vehicle Concepts, German Aerospace Center, 2025, "Next Generation Car," https://www.dlr.de/en/fk/research-and-transfer/projects/global-projects/next-generation-car, Accessed May 15, 2025.