Research in Engineering Design (2022) 33:307–349
https://doi.org/10.1007/s00163-022-00389-w

ORIGINAL PAPER

Organizing the fragmented landscape of multidisciplinary product development: a mapping of approaches, processes, methods and tools from the scientific literature

Julia Guérineau[1,2] · Matthieu Bricogne[1] · Louis Rivest[2] · Alexandre Durupt[1]

Received: 27 April 2021 / Revised: 19 February 2022 / Accepted: 3 April 2022 / Published online: 17 May 2022

Abstract
The fourth industrial revolution is shaping a new industrial landscape. A variety of technologies related to software, information and communication technologies embody a ubiquitous digital and connectivity era. These technologies enable the creation of new products with the integration of connectivity, data collection and processing capacities which require combining engineering disciplines. Increasing product multidisciplinarity compels companies to adapt their product development practices. The scientific literature offers a variety of concepts and techniques to support multidisciplinary product development. This paper seeks to organize the landscape of concepts and techniques available for multidisciplinary product development. An extensive literature review was conducted, and 236 concepts and techniques were identified. Multidisciplinary products of interest deal with both software and hardware development and can be encountered through the denominations of cyber-physical systems, mechatronics and smart products and systems. An in-depth analysis led to the classification and mapping, for each product denomination, of the concepts and techniques available to support their development. The classification relies on a four-level model paired with a decision tree to thoroughly sort the variety of concepts and techniques into the approach, process, method, and tool levels. The mapping between the sorted concepts and techniques enabled the generation of graphical representations called cartographies. These cartographies serve to support companies' transformation towards the fourth industrial revolution from the product development perspective by giving a general overview of the related literature, and guiding them in the identification of the most suitable approaches, processes, methods and tools.

Keywords Multidisciplinary product development · Mechatronics · Cyber-physical systems · Smart products · Industrie 4.0 · Literature review

1 Context: ubiquitous connectivity and the digital era shaping new products

The technological improvements in information and communication technologies (ICT) (Feng et al. 2018; Li 2017), as well as the integration and expansion of connectivity (Porter and Heppelmann 2014) and the miniaturization and lower costs of electronic components (Tomizuka 2006; Thebault 2013; Lasi et al. 2014) offer new possibilities to companies developing products. Products are now able to connect to numerous devices, to collect data within their surrounding environment, or even to act on it. There is an ever-increasing number of connected devices all around the world (Iansiti and Lakhani 2014) supported by an increasing number of connectivity technologies for short, medium and long range applications with low, medium and high speed data rates (Jara et al. 2013; Salman and Jain 2016). These changes embody the idea of a product evolution based on major technological improvements. This technology-driven product evolution has been discussed by several authors (Abramovici 2015; Isermann 2002; Thebault 2013; Zheng et al. 2018). This evolution can be represented via a timeline presenting the historical development from mechanical to mechatronic systems, through electrical and then electronic systems and

✉ Matthieu Bricogne
matthieu.bricogne@utc.fr

[1] Université de Technologie de Compiègne, Roberval, Centre de recherche Royallieu, CS 60 319, 60 203 Compiègne Cedex, France

[2] École de Technologie Supérieure de Montréal, 1100 Rue Notre-Dame Ouest, Montréal, QC H3C 1K3, Canada

software integration (Isermann 2002). These mechatronic systems, and products in a general manner, have been progressively enriched by an increasing proportion of software (Fricker 2012; Ebert 2013; Porter and Heppelmann 2014; Bricogne et al. 2016; Paluch et al. 2019). This combination of the development of information technologies (IT) and software on one side; and the integration and expansion of connectivity on the other side, have led to the emergence of a new era, qualified in this paper as the digital and connectivity era. This era establishes a new paradigm in which digital and connectivity are becoming pervasive and are shaping a new stage of product evolution. Accordingly, the present stage of product evolution is related to products integrating software and hardware with an emphasis on connectivity integration and wide-spread networks. These products are thus inherently multidisciplinary. In this paper, 'multidisciplinary products' refer to products integrating mechanics, electronics, electricity, software and connectivity. In the scientific literature, mechatronic, cyber-physical systems (CPS) and the so-called "smart" products can combine these different disciplines. In addition, both the terms "product" and "system" can be encountered. In this paper, the term "product" is preferred over "system", as the term product can be more generic and covers the spectrum "from single components to extremely complex systems" (Abramovici 2015). The upper part of Fig. 1 represents the described product evolution with the progressive integration of new technologies, allowing new capabilities and possibilities.

1.1 Industrial revolutions feed product evolution

Over decades, products were first mechanical-based and were (possibly) enriched successively by electricity, control, electronics, software and now connectivity, thus opening new possibilities. These stages could also be related to the different industrial revolutions, induced by technical breakthroughs (Drath and Horch 2014) and leading to the emergence of subsequent eras. Indeed, a revolution implies fast-paced and radical changes, whereas the subsequent era lasts for a substantial time span with the establishment of a new paradigm. According to Bloem et al., the first industrial revolution and subsequent era relied on steam-powered mechanical production and ended early in the twentieth century with the emergence of the second industrial revolution based on electricity-powered mass production, setting up another era, but one still related to hardware. The third industrial revolution started in the 70's with production automation based on electronics and IT (Bloem et al. 2014), establishing the emergence of a software era. Up to the third industrial revolution, the previously described product evolution stages and the industrial eras were in accordance and based on the same technical breakthroughs. The fourth industrial revolution is certainly no exception and is based on its own set of technological breakthroughs. This revolution is often referred to as "Industrie 4.0" (Drath and Horch 2014; Lasi et al. 2014).

Industrie 4.0 is a German initiative created in 2011 and is probably the most well known (Kagermann et al. 2013) of

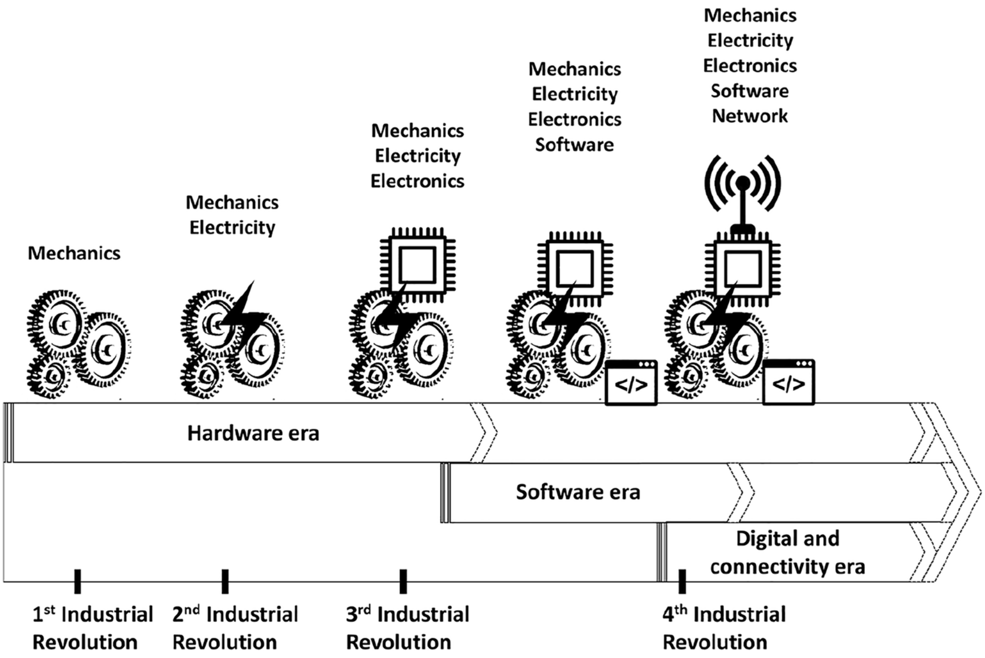
Mechanics
Mechanics Electricity
Mechanics Electricity Electronics
Mechanics Electricity Electronics Software
Mechanics Electricity Electronics Software Network
</>
</>
Hardware era
Software era
Digital and connectivity era
1st Industrial Revolution
2nd Industrial Revolution
3rd Industrial Revolution
4th Industrial Revolution

Fig. 1 Product evolution and industrial revolutions share similar technological advances

the various governmental programs and initiatives seeking to support this revolution through the establishment of roadmaps and the identification of the key technologies. In that sense, the fourth industrial revolution is more likely relying on a set of technologies that are difficult to determine at the moment due to the revolution's predictive form (Drath and Horch 2014). In that sense, Rüßmann et al., from BCG consulting, list nine transforming technologies (Rüßmann et al. 2015), Santos et al. focus on eight key technologies (Santos et al. 2017), and Danjou et al. propose a set of ten technological groups to build up an Industrie 4.0 strategy (Danjou et al. 2017). These different technologies presented by different groups of researchers are positioned as building blocks for the implementation of Industrie 4.0. From our perspective, digital—encompassing the processing, analysis, use and valorisation of data, and connectivity—networks and data exchange—are the common denominator of these various building blocks, confirming the position of the fourth industrial revolution within the digital and connectivity era.

Figure 1 shows how industrial revolutions, their subsequent industrial eras and product evolution stages are interwoven with new technologies supported by their disciplines. Pervasive connectivity and networks are envisioned as technologies for the new product generation and the fourth industrial revolution.

1.2 Reconsidering product development in the fourth industrial revolution/era

For each industrial revolution, the impact of technology transforms manufacturing and productivity (Drath and Horch 2014). In the case of the fourth industrial revolution, production systems become more autonomous and flexible, which should lead to higher productivity, quality and possibly a reduction of costs (Moeuf et al. 2018). However, as discussed previously, the digital and connectivity era is also shaping new products. Indeed, the whole product lifecycle is expected to be impacted with changes in marketing with new business models (Anderl 2014; Ibarra et al. 2018), production (Erol et al. 2016) and development.

The integration of connectivity is impacting products and their development in a number of ways. Connectivity expands the boundaries of products and offers new possibilities. Released products are now part of a larger system as the system's boundaries are expanded; products can interoperate with third-party systems and external stakeholders, some of which will emerge, evolve, be replaced or terminated along the product lifecycle. This integration is leading to an unknown and unstable environment and adds another dimension of complexity. Therefore, interoperability and the respect of standards could be key factors of successful product integration and ensure products' data exchange with their environment. A connected product can also be a part of or managed by a higher system's level, or it could control sub-systems. This approach could lead companies towards a systemic way of approaching product development and to adopt a system of systems mindset (Maier 1998). Connectivity integration also raises cybersecurity concerns and possibly ethical issues which must be considered within product development (Nyman and Främling 2008; Hughes and Cybenko 2014; O'Halloran et al. 2018).

Accordingly, products are becoming more complex, and so is their development (Kaul et al. 2017; Mabrouk et al. 2018; Zheng et al. 2019). Traditional ways of developing products may not be suitable to face the new challenges (Mabrouk et al. 2018) or cope with the variety of disciplines involved. Given this turning point, companies would do well to reconsider product development to take full advantage of the fourth industrial revolution (Herzog and Bender 2017; Rauch et al. 2016; Schuh, Rudolf, & Riesener, 2016). Indeed, the existence of a link between the product to be developed and its development was underscored by Clarkson and Eckert, who consider that product development should be tailored to the product under development (Clarkson and Eckert 2005). Furthermore, the necessity of processes, methods and tools to structure and support various aspects of product development has been acknowledged by several experts (Nijssen and Frambach 2000; Cagan and Vogel 2002; Lindemann 2003; Stetter and Lindemann 2005; Gendron et al. 2011; Albers et al. 2014). As products evolve, product development structure is expected to evolve. Guérineau et al. have proposed a four-level model that structures product development. Their four levels are approach, process, method and tool, and are defined as follows (Guérineau et al. 2018).

> The approach can be envisioned as an overall philosophy. It is a set of principles that allows product development to be addressed at a macroscopic scale. For approach operationalization, principles can be transposed in the form of processes, methods and tools.
> The process gathers a series of temporally organized mesoscopic steps—ordered sequentially or concurrently—to fulfil a purpose; these steps include the input elements (customer's requirements, specifications, etc.) and resources (financial, human, IT, etc.) required to obtain a result that could take the form of a product. A process can be iterative and can integrate established milestones. It organizes the product development steps and relies on methods and tools for steps operationalization.
> The method is a set of rules and engineering practices implemented within a process, thereby allowing a technical procedure to be realized and so to reach a result. Method's tasks can be realized using tools. Any method is part of a more general process.

> The tool is where a delineated method's task to obtain and/or improve a result is realized by acting on a particular element. A tool is thus able to support or help in the realization of a task. A tool intercedes for a defined purpose and at a specific moment of a product's development. The term tool includes editing tools to create/produce/improve a result, and management tools designed to maintain a result/a state. Tools can be fully or partially automated.

The collection of approaches, processes, methods, and tools proposed by the scientific literature is referred to as *concepts and techniques* in this paper. Moreover, this work relies on the postulate that product development requires what is called a *set* of concepts and techniques, connecting at least an approach, a process, and one or several methods and tools. A connection between the different levels is mandatory to denote a *set*. A *set* is named after the approach level. For example, an Agile *set* would be composed of Agile as an approach, paired with Scrum as a process, and be supported by a method such as continuous integration and tools such as backlogs and user stories.

Figure 2 illustrates the hierarchical and logical links in between these four levels. The four-level model will help to organize the different concepts and techniques and to highlight the links between them across the levels, as well as between existing *sets*.

Consequently, moving towards the digital and connectivity era, companies developing traditional products need to identify concepts and techniques to deploy at the different levels—the *sets*—that will support their products' evolution towards multidisciplinary products.

This paper seeks to answer the following research questions:

> What are the approaches, processes, methods and tools recommended by the scientific literature for multidisciplinary product development?
> How do the approaches, processes, methods and tools recommended for mechatronic, CPS and smart development differ?
> Do the works in the scientific literature allow to structure approaches, processes, methods and tools into consistent sets for multidisciplinary product development?

To address these research questions, this paper proposes a comprehensive view of product development through the cartography of the approaches, processes, methods and tools that can be deployed for multidisciplinary product development (MPD). This paper does not provide a selection process of the suitable concepts and techniques to deploy for a specific MPD.

Some studies have already been conducted on the analysis of the literature regarding product development (Dubberly 2004; Gericke and Blessing 2012; Hehenberger et al. 2016; Khaitan and McCalley 2015; Tomiyama et al. 2009; Zheng et al. 2014). In contrast, our study encompasses mechatronics, CPS and smart products and their respective recommended concepts and techniques. The proposed cartographies are the only ones, to the authors' knowledge,

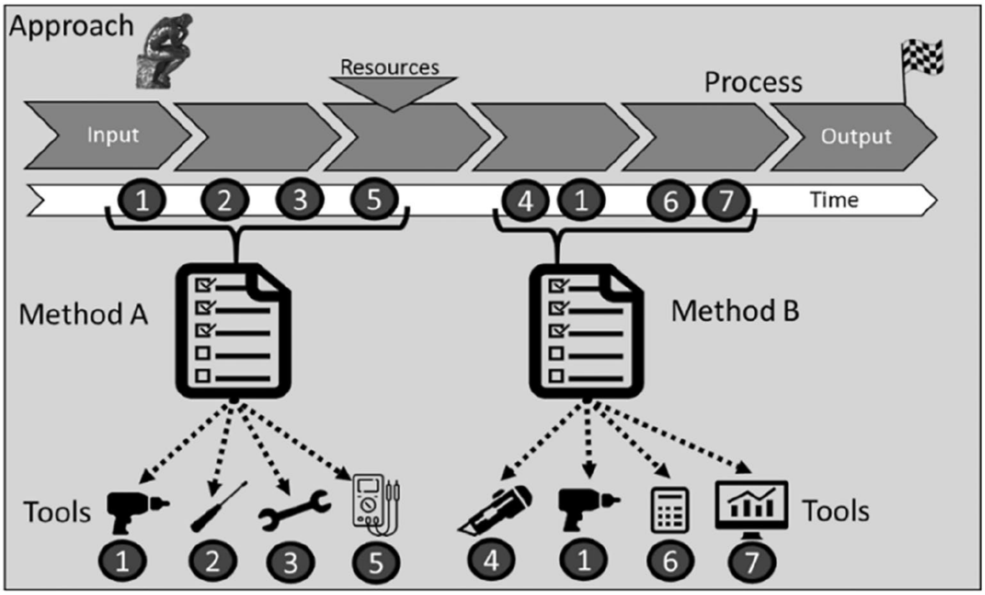

Fig. 2 The four-level model for the structuration and cartography of concepts and techniques, their hierarchy and respective links to support product development: approach, process, method and tool levels. Adapted from (Guérineau et al. 2018)

to organize 236 concepts and techniques connected by 182 links and contributing to multidisciplinary product development, as described in 167 scientific papers, and thus to help illuminate and navigate the available body of knowledge.

The cartography organizes the different concepts and techniques and their hierarchical layout based on the four-level model introduced above. The analysis is structured along two axes. The first axis is the four-level model with the defined levels to classify the variety of concepts and techniques, and the second axis is related to the type of multidisciplinary product. Hence, the four-level model will be instantiated three times, once for each type of multidisciplinary product: mechatronic products, CPS and smart products.

The following section presents the methodology adopted to perform the literature review and its analysis. The third section highlights the concepts and techniques recommended for MPD. Further analysis of the three cartographies and the research perspectives of MPD are discussed in the fourth section. The paper ends with some conclusions and recommendations for future work.

2 Methodology: a review of the concepts and techniques for multidisciplinary product development

This section details the methodology adopted for the collection and analysis of the appropriate scientific literature, as well as for the representation of the results.

2.1 Collecting concepts and techniques for multidisciplinary product development

This paper focuses on the literature related to mechatronics, CPS and smart product development. To conduct the literature review, we first needed to identify the databases and appropriate keywords. In the definition of our research scope, certain elements were excluded. Accordingly, our work excludes computer-based tools such as computer-aided design, product data management, software configuration management, integrated development environment, simulation tools and simulation data management, which can be used in the development process. Indeed, these computer-based tools are considered as an agnostic support to the approach, process, methods and tools implemented for MPD.

To address the different spellings and combinations of words, the following three queries were formulated "Mechatronic* product*" OR "Mechatronic* system*"; "Cyber physical system*" OR "Cyber-physical system*"; and "Smart product*" OR "Smart system*". In addition, each query was completed with "development" OR "design". Indeed, design is part of the overall development (Roozenburg and Eekels 1995; Vajna et al. 2005; Ulrich and Eppinger 2016), and product development can encompass conceptual design, embodiment design and detail design (Pahl et al. 2007; Gericke and Blessing 2012). The queries were submitted to Scopus and Web of Science. As the core of the papers must be related to the product development topic, the queries were limited to the titles. In addition, the queries were also limited to journal articles.

The database results were merged to delete duplicates and compose an initial list of 304 articles: 161 in mechatronics, 105 on CPS and 38 on smart products. This list was filtered to establish a list of relevant articles. To be considered as relevant, an article must focus on development—or design—and integrate concepts and techniques considerations. However, some elements retrieved from the queries were excluded from our scope of work. Although simulation and optimization techniques, such as finite elements, as well as testing are part of product development, they are not analyzed in-depth here. Indeed, while simulation and optimization techniques can be based on design models, which are discussed through modelling, design optimization and simulation are considered as another facet of the literature. Hence, this study is limited to the design models and not their further exploitation to conduct simulations, optimization or testing.

Finally, in tandem with the list established from the databases, we conducted backward snowballing sampling. Backward snowballing can help to complement a corpus with additional references (Wohlin 2014; Jalali and Wohlin 2012). The resulting list of articles allowed us to extract a collection of concepts and techniques for each type of product. These concepts and techniques were classified according to the decision tree presented in the next section.

2.2 Analyzing and sorting the variety of concepts and techniques

To analyze the relevant articles, each of the concepts and techniques collected was classified and represented within the four-level model, either as an approach, a process, method, or tool. To conduct this classification, a decision-tree was designed based on the definitions provided by Guérineau et al. (2018), illustrated in Fig. 3. The classification relies on three questions related to the purpose, the characteristics and a frequent feature found in each class. The purpose addresses what the concept or the technique is used for, or what it allows. The characteristics are related to the intrinsic nature of the concept or technique, and under which form they are usually encountered. The frequent feature represents an additional feature that is usual but not mandatory. This is an additional validation of the two previous questions.

To ensure an accurate classification, at least a definition or a short presentation of the usage of the concepts and

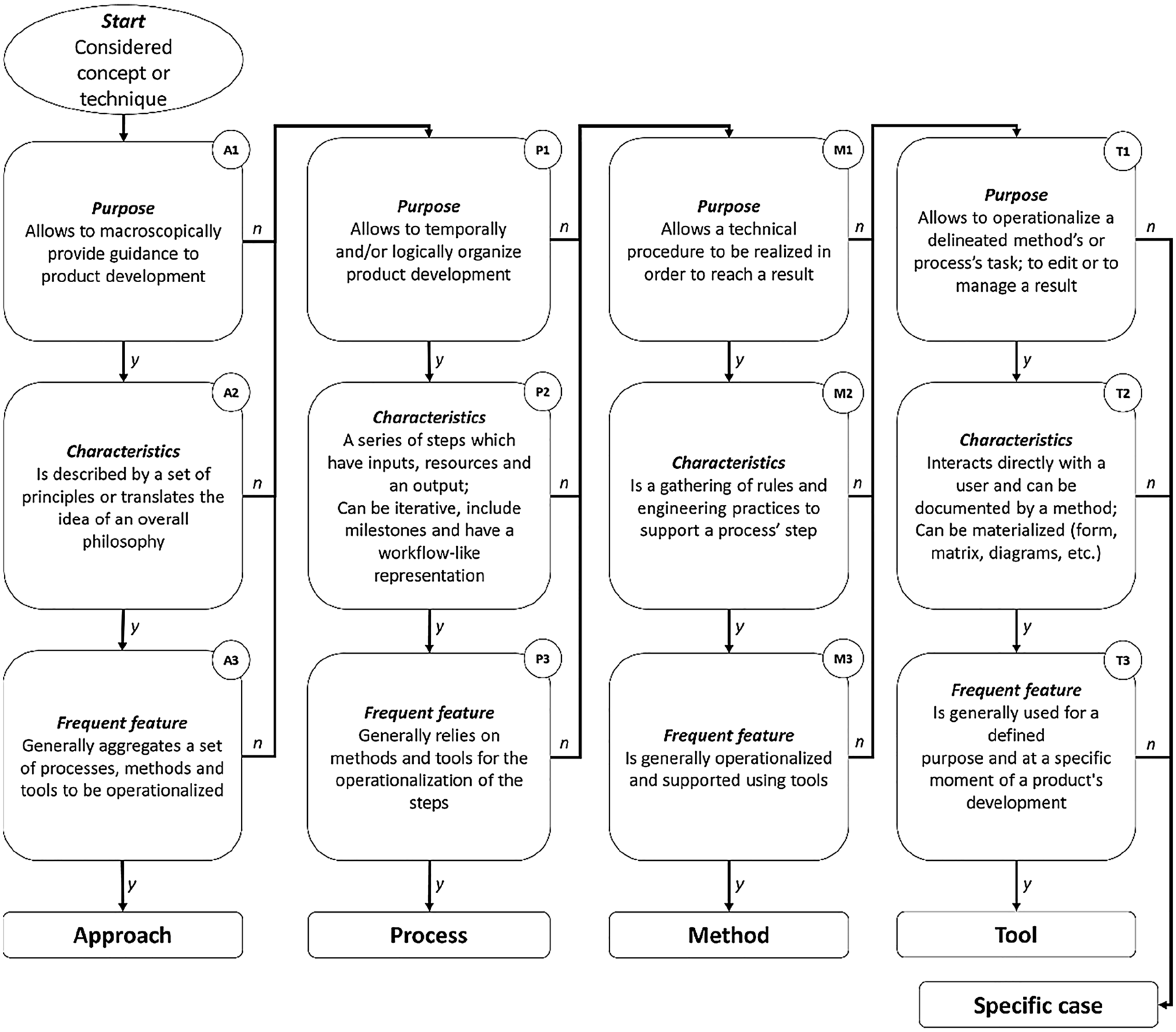
Start
Considered concept or technique
A1
Purpose
Allows to macroscopically provide guidance to product development
A2
Characteristics
Is described by a set of principles or translates the idea of an overall philosophy
A3
Frequent feature
Generally aggregates a set of processes, methods and tools to be operationalized
Approach
P1
Purpose
Allows to temporally and/or logically organize product development
P2
Characteristics
A series of steps which have inputs, resources and an output;
Can be iterative, include milestones and have a workflow-like representation
P3
Frequent feature
Generally relies on methods and tools for the operationalization of the steps
Process
M1
Purpose
Allows a technical procedure to be realized in order to reach a result
M2
Characteristics
Is a gathering of rules and engineering practices to support a process' step
M3
Frequent feature
Is generally operationalized and supported using tools
Method
T1
Purpose
Allows to operationalize a delineated method's or process's task; to edit or to manage a result
T2
Characteristics
Interacts directly with a user and can be documented by a method;
Can be materialized (form, matrix, diagrams, etc.)
T3
Frequent feature
Is generally used for a defined purpose and at a specific moment of a product's development
Tool
Specific case
y
n

Fig. 3 Decision tree for systematic classification of concepts and techniques within the four-level model

techniques is required from each journal article. Taking Agile and the information provided by the Agile Manifesto (Beck et al. 2001) as an example, such a decision tree could be used as follows (see the first column of Fig. 3). For the A1 box, Agile can be envisioned as high-level and is related to software product development; the A2 box is supported by the 12 principles of the Manifesto and the underlying Agile philosophy; and finally, the A3 box is supported by the existence of SCRUM, Extreme Programming and other concepts and techniques that operationalize Agile. Accordingly, Agile is classified an approach. As for SCRUM, based on the information provided by Schwaber (1997), the A1 box can be answered positively, but not A2, as it is not a set of principles—despite a list of characteristics—, nor a philosophy itself. Instead, SCRUM organizes the product development following "phases" and "steps" towards the development of a product, thus validating the P1 and P2 boxes. For P3, Schwaber mentions relying on "Object-oriented techniques". Therefore, SCRUM is a process.

In practice, different authors can define a concept or a technique with different definitions and hence have a different vision or opinion regarding its use. Three options are possible for these specific cases, considered in the following sequence of priority. The first option relies on the definition considered as initial in the scientific literature, the second option analyzes the definition referenced by the paper studied in this literature review, and the third option is to find a common agreement and present the different views with their appropriate references. For each specific case, the option adopted in this work is so noted.

2.3 Representing the results with cartographies

Relying on the four-level model, a “cartography” is the graphical representation adopted to organize the fragmented landscape of multidisciplinary product development concepts and techniques. Moreover, the cartographies—one for each type of multidisciplinary product—enable an advanced stage of analysis. Indeed, graphical representation is a way to organize complex information and thus should facilitate the knowledge transfer of product development practices.

The adopted representation of the analysis is codified with graphical elements. The graph legend is illustrated in Fig. 4. For a clearer representation, the concepts and techniques are dispatched on different figures. If a concept or technique is reused across different figures, the font will be grey, bold, and italic. A box with a continuous border stands for a concept or a technique expressly cited by the scientific

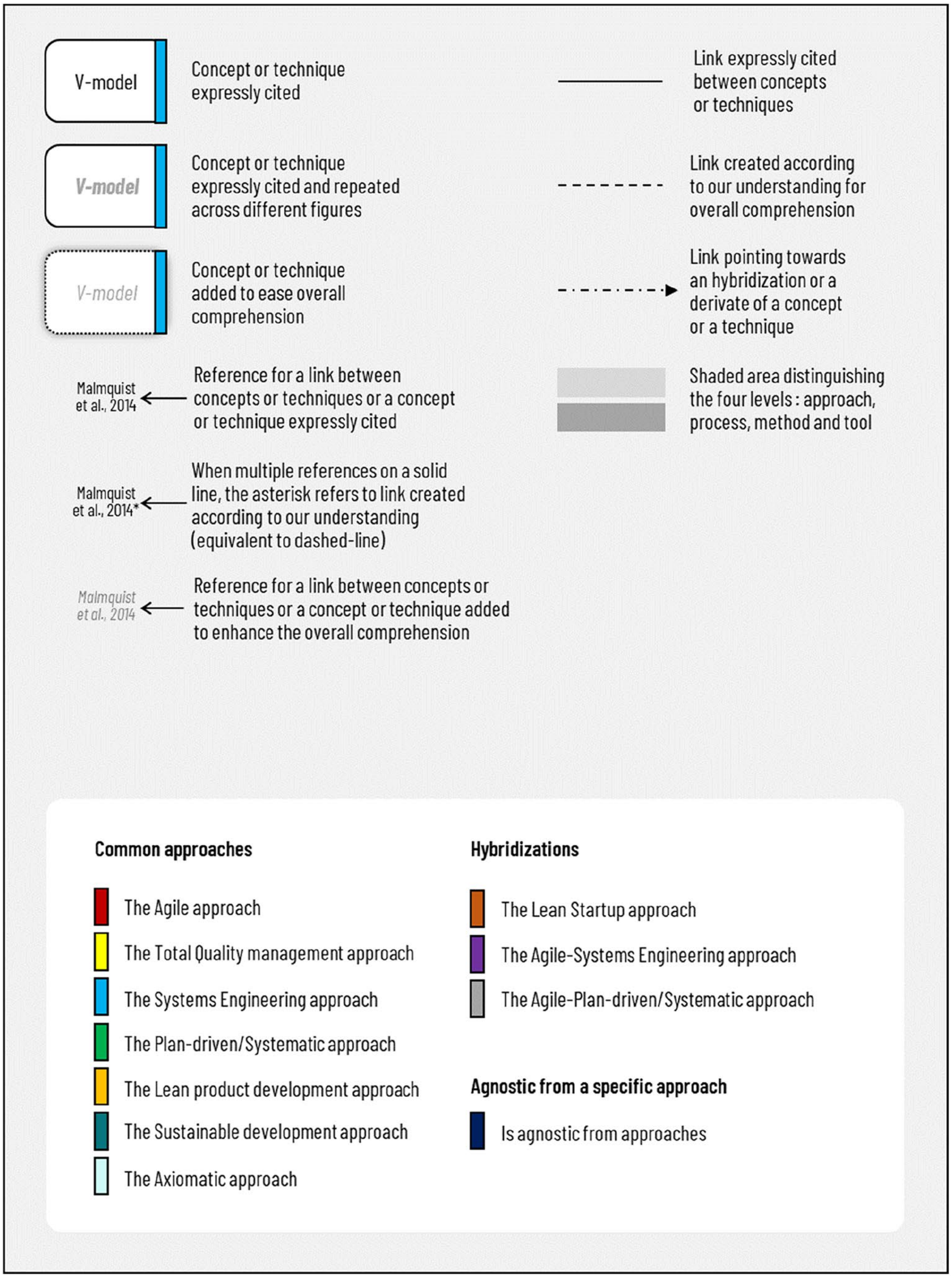
V-model
Concept or technique expressly cited
V-model
Concept or technique expressly cited and repeated across different figures
V-model
Concept or technique added to ease overall comprehension
Malmquist et al., 2014
Reference for a link between concepts or techniques or a concept or technique expressly cited
Malmquist et al., 2014*
When multiple references on a solid line, the asterisk refers to link created according to our understanding (equivalent to dashed-line)
Malmquist et al., 2014
Reference for a link between concepts or techniques or a concept or technique added to enhance the overall comprehension
Link expressly cited between concepts or techniques
Link created according to our understanding for overall comprehension
Link pointing towards an hybridization or a derivate of a concept or a technique
Shaded area distinguishing the four levels : approach, process, method and tool
Common approaches
The Agile approach
The Total Quality management approach
The Systems Engineering approach
The Plan-driven/Systematic approach
The Lean product development approach
The Sustainable development approach
The Axiomatic approach
Hybridizations
The Lean Startup approach
The Agile-Systems Engineering approach
The Agile-Plan-driven/Systematic approach
Agnostic from a specific approach
Is agnostic from approaches

Fig. 4 Cartography legend and color code for the identification of similar product development approaches

literature and is paired with the cited reference(s). A box with a dotted border indicates a concept or a technique added by the authors of this survey to facilitate the overall comprehension. This is especially useful in the case of hybridizations or in the case of a derived concept or technique, as described in the next paragraph. These additions also help to specify the approach level to identify a philosophy and the matching colored rectangle.

A single dash-dotted arrow pointing to a concept or technique indicates a derivation. A derivation is an adaptation, or specialization, of a concept or a technique to specific purposes, and accordingly relies on an existing foundation. For instance, the W-model derives from the V-model (Nattermann and Anderl 2010). Two dash-dotted arrows indicate a hybridization, and the origins of the arrows are the originating concepts or techniques. A hybridization emerges from a combination of two (or more) concepts and techniques whose resulting value is expected to be much higher than if they were utilized separately (Guérineau, Rivest, Bricogne, & Durupt, 2016).

The references of an expressly cited concept or technique are colored in black, as opposed to the light grey colored references that are mentioned to enhance the cartography by adding a link, a concept or a technique but do not address MPD. If a reference appears along a solid line, it means that both concepts and techniques were cited and linked together. An asterisk indicates that the reference on the link is interpreted by the authors and is equivalent to a dashed line. Hence, the dashed line is a link created according to our understanding of the relations between concepts and techniques, indicating a specific reference (or not). Finally, the shaded areas help to distinguish the four levels of the model. These details are illustrated in Fig. 4.

In addition to the graphical codification, the colored rectangle on the right side of the boxes help to identify similar approaches and their operationalization. Each approach has its own color, which in turn helps to identify possible *sets*. The different colors and associated approaches are grouped by category at the bottom of Fig. 4.

3 Results: cartographies of concepts and techniques for multidisciplinary product development

After presenting the methodology and the graphical representation convention of the analysis, this section presents the results of our analysis of the concepts and techniques for MPD found in the scientific literature. Sections 3.1, 3.3 and 3.5 list and briefly review the concepts and techniques for mechatronics, CPS and smart product development, respectively, to be represented in their unique cartography. The concepts and techniques presented in each of these sections are positioned based on their authors' self-identification regarding mechatronics, CPS or smart product development. Sections 3.2, 3.4 and 3.6 introduce the cartography of concepts and techniques for mechatronics, CPS and smart product development, respectively. A searchable version of these cartographies, which makes it much easier to pinpoint references, concepts, and techniques is available in (Guérineau et al. 2022). Section 3.7 proposes a synthesis and uses metrics to substantiate the conclusions drawn across the different sections. In addition, the application of the cartography's construction procedure is exemplified throughout Sect. 3.1.

3.1 Concepts and techniques for mechatronic product development

Mechatronics appeared in Japan in the 1960's and initially referred to a proprietary name qualifying products that integrate mechanics and electronics (Harashima et al. 1996). Today, only products that integrate these two disciplines would be qualified as electromechanical products. The concept of mechatronics has evolved and is now distinct from electromechanics. Mechatronics has been enriched with computer science (Harashima et al. 1996; Turki et al. 2005; Abramovici and Bellalouna 2007; Freddi 2009), and "control engineering" is sometimes mentioned as being part of mechatronics (Craig and Stolfi 1994; AFNOR 2008; Penas et al. 2010; Warniez et al. 2012). Pannaga, Ganesh, & Gupta (2013) extend the definition to hydraulics, pneumatics and optics, and Tomizuka (2006) proposed a generic architecture of a modern mechatronic system that includes connectivity. The integration of software has opened new possibilities for function implementation (Bricogne et al. 2016). Combining these disciplines and the positive interactions that can emerge thereof is making the development of mechatronics complex (Hehenberger and Bradley 2016). This complexity and the tight links between components requires the adoption of new concepts and techniques (Lapusan et al. 2010). The recommended concepts and techniques for mechatronic product development are discussed next. Sections 3.1.1 and 3.1.2 discuss systems engineering and the V-model, respectively, along with their related concepts and techniques. Section 3.1.3 shows how the cartography is built based on the information provided in the previous sub-sections.

Sections 3.1.4 and 3.1.5 present works related to model-based and modelling techniques. Section 3.1.6 discusses modular design among other techniques, such as design structure matrix and quality function deployment. Sections 3.1.7, 3.1.8, 3.1.9, and 3.1.10 focus on four different approaches and their related concepts and techniques: Agile, Systematic design, Axiomatic design and Eco-design, respectively. Section 3.1.11 presents integrated product development, and 3.1.12 discusses other concepts and techniques.

3.1.1 Systems engineering and related concepts and techniques

Systems engineering (SE) is an acknowledged approach to support mechatronic product development and is generally associated with the V-model (Dieterle 2005; Kleiner and Kramer 2013; Mcharek et al. 2019; Mhenni et al. 2014; Rothful et al. 2006; Sünnetcioglu et al. 2016; Turki et al. 2005; Zheng et al. 2017). This association between SE and initially the "Vee process model" was coined by Forsberg and Mooz (1991); however, they did not address mechatronic product development. Later researchers have completed this association with other concepts and techniques.

Accordingly, Mcharek et al. introduced knowledge management concerns to ease collaboration throughout the product development process, implemented via the KARREN platform (Mcharek et al. 2018, 2019). From our perspective, knowledge-based engineering is considered as a method, given that the reuse of knowledge can be considered as an engineering practice. This aspect of knowledge is also adressed by Delbecq et al. (2017), who envision the possibility, through modelling, of building component and system models that can be stored and reused. Their proposed framework supports model-based design and integrates both acausal and causal modelling. In addition, their work notes that modelling supports SE processes—the "systems engineering process" block.

Other authors have complemented SE and V-model association with modelling techniques, including Function-Behavior-State modelling (Habib and Komoto 2014; Komoto and Tomiyama, 2010, 2012), model-based design (Dieterle 2005; Kleiner and Kramer 2013), and system modelling techniques (Rothful et al. 2006). Within the SE and V-model association, Mhenni et al. (2014) focus on the system design step. The authors propose a SysML-based "methodology" for system modelling and describe the use of system modelling techniques through the different SysML diagrams. Beyond the use of system modelling techniques, their "methodology" is structured by two procedures, a black box analysis and a white box analysis, and is thus considered as a method. From our understanding, this method is complemented by a functional breakdown and analysis that can be performed using APplication aux Techniques d'Entreprise (APTE), the Function analysis system technique (FAST) diagram, or Structured Analysis and Design Technics (SADT). Also from our understanding, system modelling techniques are utilized to support the functional breakdown and analysis.

Couturier et al. position their work within an SE approach and the V-model, complemented by model-based systems engineering (MBSE)—the "model-based and model-driven practices" block. The V-model is envisioned as iterative, and several passes are necessary to obtain a mature product. The authors also discuss the use of axiomatic design, Function-Behavior-State and Function-Behavior-Structure for reasoning, synthesis, analysis and evaluation purposes (Couturier et al. 2014). Function-Behavior-State was initially defined in (Umeda et al. 1996).

Within an SE approach, Zheng et al. (2017) also discuss an adapted V-model as a macro-level design process paired with a hierarchical design model as a micro-level design process (Zheng et al. 2017). From our understanding, the hierarchical design model integrates an underlying engineering practice of hierarchical decomposition supporting the hierarchical architecture at each design step of the V-model's downward side. Therefore, the hierarchical design model can be envisioned as a method. Moreover, the multidisciplinary interface modelling technique, integrating an interface model, is introduced to ensure consistency between the two levels of the design "process". Depending on its usage, the interface modelling technique can be argued as a method or as a tool. Indeed, if ensuring consistency is considered a delineated task, the technique will be positioned as a tool. On the other hand, if it is considered as an engineering practice supported by the interface classification and the interface compatibility check, the interface modelling technique will be positioned as a method (Zheng et al. 2016; Zheng et al. 2019). While both visions are acceptable, the second vision is adopted due to its relative positioning with tools. Thus, the interface modelling technique is considered as a method. Also linked with the use of the interface modelling technique, especially its associated interface model, Zheng et al. (2019) propose the use of an "interface model-based configuration design method" for configuration design purposes, based on modularization and the use of the developed interface model to verify the compatibility of the designed modules. Accordingly, the authors introduce a link with modular design through this modularization.

The concepts and techniques discussed in this subsection are represented in Figs. 6 and 8.

3.1.2 V-model and related concepts and techniques

The V-model—also encountered as the V-cycle—is an acknowledged process for mechatronic product development. The VDI2206 is commonly cited (Malmquist et al. 2014) and adapts the V-model to the development of mechatronic systems by combining mechanical and electronic engineering with information technology, all supported by modelling and model analysis (VDI-Fachbereich Produktentwicklung und Mechatronik 2004). Malmquist et al. discuss the VDI2206's V-model and its iterative form. They also propose a "holistic design methodology" that promotes the concurrent development of mechanical, electrical and control domains and which relies on building models from component libraries (Malmquist et al. 2014). From our understanding, the authors use a model-based design—“model-based and model-driven practices” block—supported by components-oriented modelling, which in turn allows to conduct the system’s optimisation.

Inspired by the VDI2206’s V-model, Oestersötebier et al. propose a design process for “intelligent mechatronic systems” (Oestersötebier et al. 2012). However, Oestersötebier et al. differ from the previous solutions by their use of Web Semantic and ontologies, which are envisioned as a way to store knowledge. This knowledge base can store sub-models and solution patterns to be reused across projects, thereby reducing the modelling efforts, supporting model-based design, and promoting the “design of intelligent mechatronic systems” design process (Oestersötebier et al. 2012). The modelling was realized thanks to object-oriented modelling (Oestersötebier et al. 2012). In line with reuse practices, Bachmann and Messnarz suggest using the V-model, and promote the reuse of software and system modules to implement a platform strategy (Bachmann and Messnarz 2012). For complementary readings on reuse-oriented software engineering, see Sommerville (2010).

The V-model can also be supported by system modelling techniques (Vasić and Lazarević 2008; Valasek 2016). Valasek discusses the design through modelling and synthesizes a view of the various modelling techniques including equations, blocks, multipoles and bond graphs (Valasek 2016).

Other authors also use the V-model to position their contribution. Accordingly, Habchi and Barthod propose a ten-step methodology in which different methods and tools are gathered and organized to support the downward side of the V-model. The methods and tools used are rooted in a mechanical design background with the use of APTE, supported by the Octopus diagram and the Horned beast, as well as SADT, FAST and functional block diagrams for functional breakdown and analysis. For their architecture, the authors make use of block diagrams, followed by petri nets for system behavior and FMEA for dysfunctional analysis (Habchi and Barthod 2016). Similarly, Hehenberger positions his contribution within the V-model, exploring the model-based parametric design and discussing hierarchical models (Hehenberger 2015). Plateaux et al. also focus on the downward part of the V-model, and make use of SADT and SysML for functional analysis (Plateaux et al. 2010). Linked with functional breakdown and analysis, Derelöv et al. (2008) propose to rely on an extension of Hubka-Eder’s model from the theory of technical systems (Hubka and Eder 1988) to realize a functional modelling.

Alvarez Cabrera et al. position their work within the V-model and propose an architectural model to support systems architecting activities. Their architectural model is, from our understanding, linked to model-based practices, and encompasses four layers: function, structure, behavior, and external communications (Alvarez Cabrera et al. 2011). This architectural model is also discussed in Hehenberger (2014), and is, from our understanding, linked with the hierarchical modelling technique. The architectural model can be positioned at the method level, as it supports the system architecture and decomposition according to a technical procedure—the layers—and aims to create and use a product representation.

Finally, Barbieri et al. suggest the use of the W-model, a derivate of the VDI 2206 V-model proposed by Nattermann and Anderl (2010), and which was initially proposed for adaptronic systems. Although adaptronics is presented by Nattermann and Anderl as being close to mechatronics and to involve similar disciplines, the reference is considered as an addition to enhance the overall comprehension of the cartography. Complementary to the W-model, Barbieri et al. also mention the use of model-based design supported by system modelling conducted in SysML, and of FMEA for its fault management strategy based on the functional breakdown and analysis (Barbieri et al. 2014).

The concepts and techniques presented above are represented in Figs. 6 and 8.

3.1.3 Exemplifying: building the cartography incorporating SE and the V-model

The proposed cartographies are rather complex. To better understand them, it is useful to explain how they were built. Based on the work introduced thus far, this subsection aims at exemplifying the transposition from the textual enumeration of concepts and techniques into the literature-based cartography. Accordingly, Fig. 5 illustrates the step-by-step procedure applied for the cartography’s construction. The cartography’s legend and color codification—see Fig. 4—are recommended to be printed separately to interpret the cartography and its construction.

Section 3.1.1 discusses SE, considered as an approach according to the decision tree. However, among the various references discussed in that section, only two address SE: Sünnetcioglu et al. (2016) and Turki et al. (2005). These references are associated with the block in Fig. 6. The rectangle is colored light blue, indicating the SE approach, as stated in Fig. 4. This first step is illustrated as Step 1 in Fig. 5.

Next, Sect. 3.1.2 reviews V-model usage for mechatronic product development. The V-model is classified as a process by the decision tree. As in Step 1 above, among the various references discussed, only one reference discusses the V-model: Malmquist et al. (2014). This reference is associated with its block in Fig. 6. In addition, despite the fact that a V-model is initially a software engineering process, its adaptations as presented by the different authors tend to

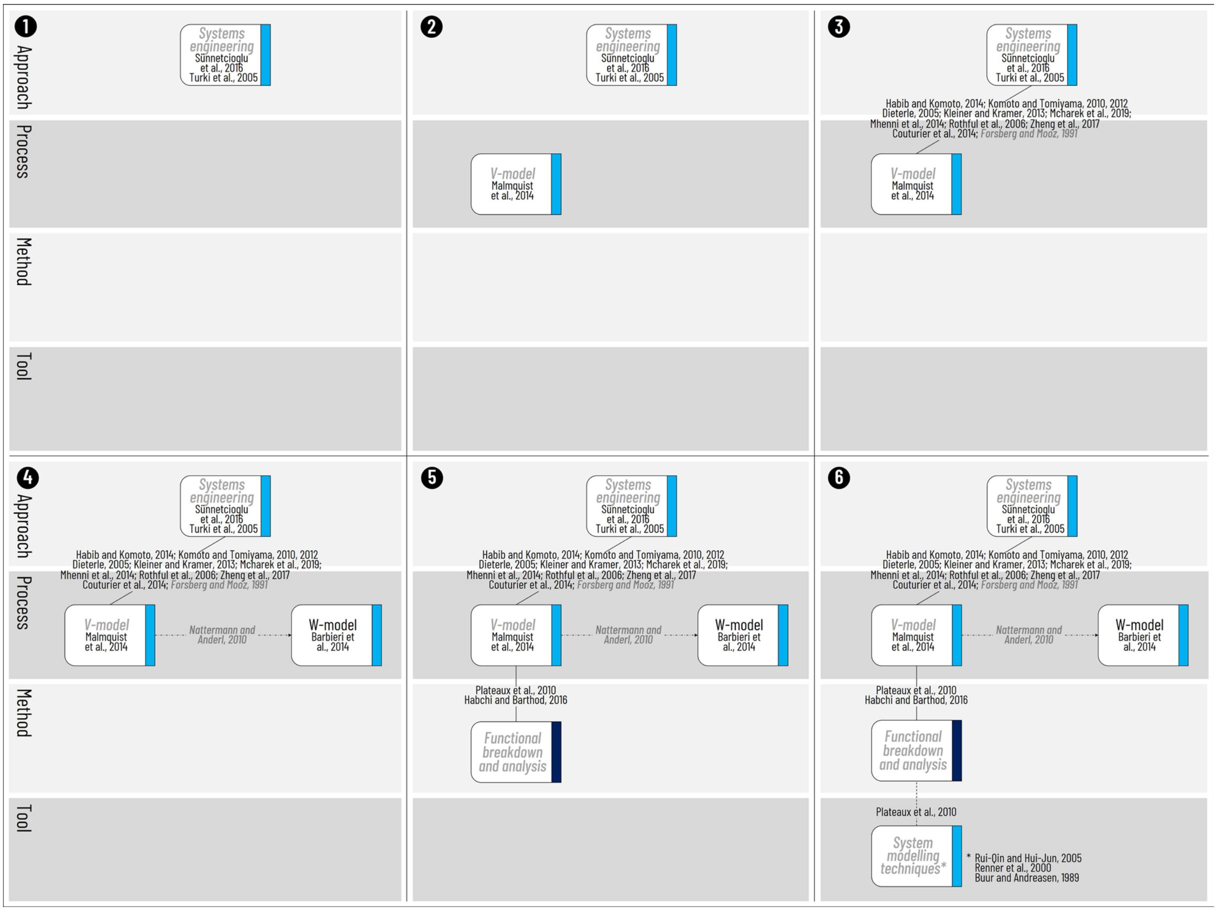
1
Approach
Process
Method
Tool
Systems engineering
Sünnetcioglu et al., 2016
Turki et al., 2005
2
Systems engineering
Sünnetcioglu et al., 2016
Turki et al., 2005
V-model
Malmquist et al., 2014
3
Systems engineering
Sünnetcioglu et al., 2016
Turki et al., 2005
Habib and Komoto, 2014; Komoto and Tomiyama, 2010, 2012
Dieterle, 2005; Kleiner and Kramer, 2013; Mcharek et al., 2019;
Mhenni et al., 2014; Rothful et al., 2006; Zheng et al., 2017
Couturier et al., 2014; Forsberg and Mooz, 1991
V-model
Malmquist et al., 2014
4
Approach
Process
Method
Tool
Systems engineering
Sünnetcioglu et al., 2016
Turki et al., 2005
Habib and Komoto, 2014; Komoto and Tomiyama, 2010, 2012
Dieterle, 2005; Kleiner and Kramer, 2013; Mcharek et al., 2019;
Mhenni et al., 2014; Rothful et al., 2006; Zheng et al., 2017
Couturier et al., 2014; Forsberg and Mooz, 1991
V-model
Malmquist et al., 2014
Nattermann and Anderl, 2010
W-model
Barbieri et al., 2014
5
Systems engineering
Sünnetcioglu et al., 2016
Turki et al., 2005
Habib and Komoto, 2014; Komoto and Tomiyama, 2010, 2012
Dieterle, 2005; Kleiner and Kramer, 2013; Mcharek et al., 2019;
Mhenni et al., 2014; Rothful et al., 2006; Zheng et al., 2017
Couturier et al., 2014; Forsberg and Mooz, 1991
V-model
Malmquist et al., 2014
Nattermann and Anderl, 2010
W-model
Barbieri et al., 2014
Plateaux et al., 2010
Habchi and Barthod, 2016
Functional breakdown and analysis
6
Systems engineering
Sünnetcioglu et al., 2016
Turki et al., 2005
Habib and Komoto, 2014; Komoto and Tomiyama, 2010, 2012
Dieterle, 2005; Kleiner and Kramer, 2013; Mcharek et al., 2019;
Mhenni et al., 2014; Rothful et al., 2006; Zheng et al., 2017
Couturier et al., 2014; Forsberg and Mooz, 1991
V-model
Malmquist et al., 2014
Nattermann and Anderl, 2010
W-model
Barbieri et al., 2014
Plateaux et al., 2010
Habchi and Barthod, 2016
Functional breakdown and analysis
Plateaux et al., 2010
System modelling techniques*
* Rui-Qin and Hui-Jun, 2005
Renner et al., 2000
Buur and Andreasen, 1989

Fig. 5 Construction example of literature-based cartography—Systems engineering and V-model

belong to the SE approach (INCOSE 2015). Therefore, the rectangle color matches that of the SE approach (light blue). This operation is illustrated in Step 2 of Fig. 5.

Furthermore, Sects. 3.1.1 and 3.1.2 presented numerous links established by different authors using SE and the V-model together. Accordingly, a solid line is drawn between SE and the V-model in Fig. 6, and endorsing references are positioned along the line. Forsberg and Mooz (1991) do not explicitly address mechatronics and are so positioned to enhance the cartography. Accordingly, that reference is in light grey and in italics, as denoted in Fig. 4. This operation is represented by Step 3, Fig. 5.

Step 4 of Fig. 5 illustrates the derivation of a concept or a technique. A derivation is depicted by a single dash-dotted arrow attached to the originating concept and technique, pointing towards the derivation. In this case, sharing similar features, the W-model proposed by Nattermann and Anderl (2010) is derived from the V-model. Since the W-model was introduced for adaptronics and not mechatronics—even though they are similar—the reference is in light grey and in italics (Fig. 6). This is represented by Step 4, Fig. 5.

Step 5 illustrates the addition of functional breakdown and analysis, proposed by Habchi and Barthod (2016) and Plateaux et al. (2010) and classified as a method. Moreover, both references expressly use functional breakdown and analysis to support the downward side of the V-model, resulting in a solid line between the V-model and functional breakdown and analysis. However, functional breakdown and analysis are rooted in mechanical engineering (Plateaux et al. 2010), and from our understanding, are not readapted to SE's constraints. Accordingly, functional breakdown and analysis are considered here as agnostic from any approaches. The rectangle is therefore colored in dark blue, as stated in Figs. 4 and shown in 6.

Finally, Step 6 adds the system modelling techniques, classified as tools. This block gathers the works that discuss miscellaneous modelling tasks or works expressly relying on the Unified Modeling Language (UML), SysML or other languages and symbolic representations. Some authors exclusively address system modelling techniques. Accordingly, these references are associated to the block (Buur and Andreasen, 1989; Renner et al. 2000; Rui-Qin and Hui-Jun 2005), as detailed in Sect. 3.1.4 and shown in

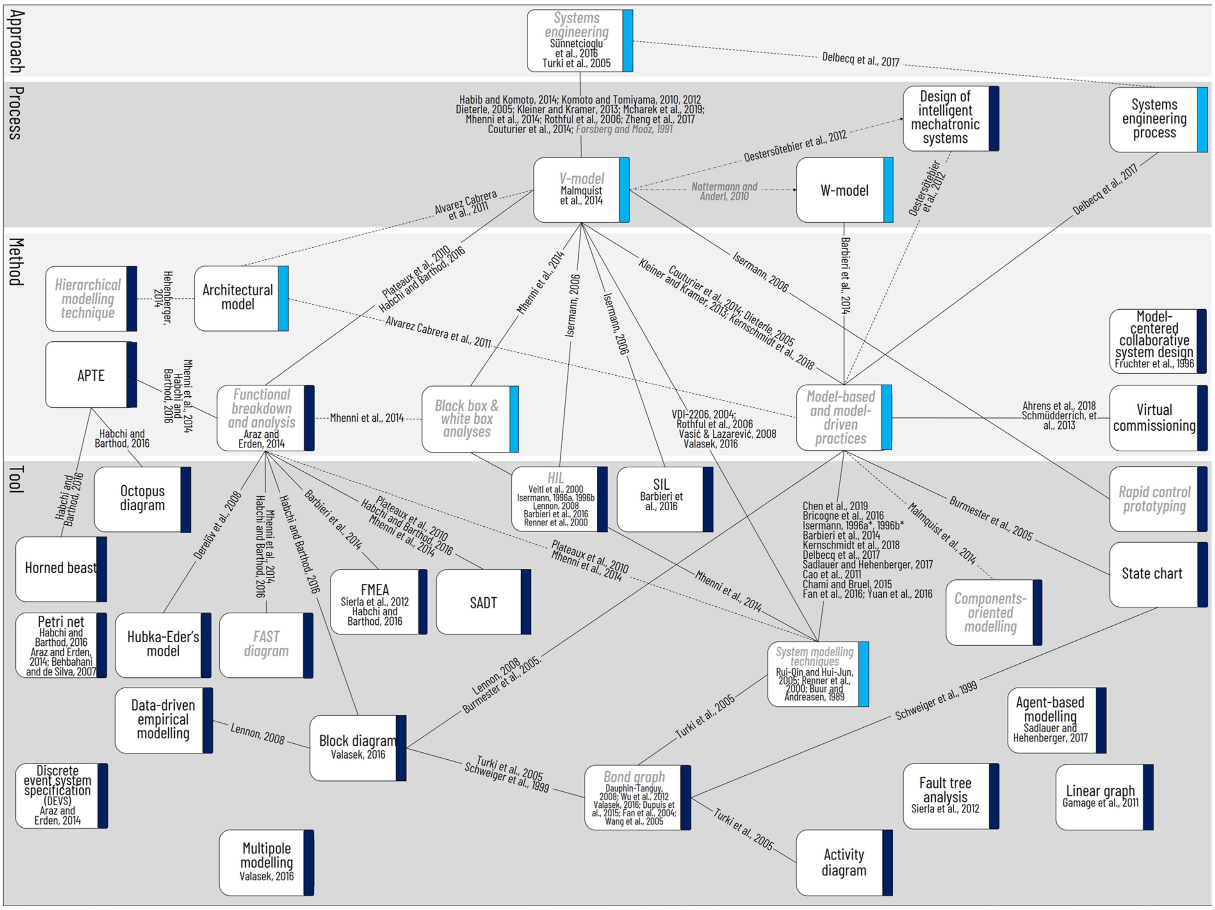

Fig. 6 Cartography of approaches, processes, methods and tools available for mechatronic product development—Focus on the SE set

Fig. 6. In addition, Plateaux et al. (2010) discuss system modelling techniques' usage. However, the link with functional breakdown and analysis is not expressly stated, and is instead interpreted. Therefore, the link with the associated reference (Plateaux et al. 2010) is drawn with a dashed line, as shown in Fig. 6.

Each subsection details an area or a branch of the cartography. At the end of each subsection, the text mentions which figures contain the concepts and techniques discussed. Accordingly, to retrieve a concept or a technique and its reference from the textual enumeration on the cartography, a first search is for the mentioned figures, then the concept or the technique in their respective blocks. The reference is positioned within or around a block or along the links emerging from that block. In addition, when a concept or a technique is repeated on multiple figures, the font is grey, bold and italic, as with systems engineering, V-model, functional breakdown and analysis, or system modelling techniques, as shown in Figs. 5 and in 6.

3.1.4 Model-based and modelling concepts and techniques

Modelling for mechatronic product development can be considered as a widespread practice. This practice was discussed in 1989 by Buur and Andreasen, who considered modelling as a way of "buying" information in the early stages, so as to cope with complexity and reduce the risk of false decisions (Buur and Andreasen 1989). They underscored the need for "a model language or a model type that could improve communication both between mechanical, electrical and software engineers, and between the project team and, for example, managers and users", hence placing models at the core of collaboration. Based on similar observations, Fruchter et al. (1996) proposed a model-centered collaborative system design with the use of form, behavior, and function models. Linked with modelling, Rui-Qin and Hui-Jun (2005) proposed their own symbolic representation to model mechatronic systems. This interest in models can be explained by the numerous possibilities to be explored. For instance, models allow to simulate and analyze a system's behavior early in the design process (Schmüdderrich et al. 2013), allowing errors to be detected earlier and making it both easier and cheaper to fix them. In that sense, some authors combine model-based design and virtual commissioning (Schmüdderrich et al. 2013; Ahrens et al. 2018). Indeed, virtual commissioning allows the creation of a copy of a real system for early testing (Ahrens et al. 2018). Schmüdderrich et al. (2013) also make use of modular design.

Modelling is also linked with the implementation of MBSE, which can be combined with the design structure matrix (DSM) to "support system architecture elaboration" (Bricogne et al. 2016). The combination of MBSE with system modelling supported by SysML language has been discussed by various authors (Bricogne et al. 2016; Cao et al. 2011; Chami and Bruel 2015; Fan et al. 2016; Yuan et al. 2016). Cao et al. state that "MBSE is the mainstream method for complex system design" as well as the fact that "MBSE facilitates dependency tracing between different models and the reuse of knowledge" (Cao et al. 2011). Additionally, on an MBSE basis, some authors focus on model verification, such as the work proposed by Chen et al. (2019), in which system design is performed in SysML—"system modelling techniques" block.

Concurrently to MBSE practices, model-based and model-driven practices can be combined with engineering, development, architecture and design. For further background on the differences between model-driven and model-based, Kernschmidt et al. (2018) offer an explanation of their divergences. These authors also propose a framework for mechatronic development positioned within a V-model, relying on model-based engineering supported by system modelling techniques conducted in SysML. They rely on the properties to compare solution alternatives and libraries to speed up the development through the reuse of modules and components (Kernschmidt et al. 2018). Nonetheless, no concept or technique related to this reuse practice is expressly mentioned. However, from our understanding, Kernschmidt et al. envisoned the reuse practices as a means to support model-based engineering. In the cartography, we mapped knowledge-based engineering with model-based practices.

Burmester et al. position their work within a model-driven architecture paradigm and suggest the use of model-driven development complemented by state chart and block diagrams (Burmester et al. 2005). Linked with modelling and model-based development, Sadlauer & Hehenberger (2017) focus on the use of general-purpose modelling languages and model-based description languages to support the early design phase of the VDI 2221 process. The authors discuss the use of DSM for dependency management and agent-based modelling as a perspective for MPD.

Alvarez Cabrera et al. (2010) discuss a variety of concepts and techniques. However, within their framework, they only explicitly mention the use of an extended Function-Behavior-State modelling, object-, component- and process-oriented techniques, and knowledge-based engineering. Within their knowledge base, they use component- and process-oriented modelling paradigms. However, they do not explain how these concepts and techniques relate to one another.

For modelling, some authors have promoted Petri Nets (Behbahani and de Silva 2007; Araz and Erden 2014). Araz and Erden propose a framework based on functional decomposition (equivalent to functional breakdown and analysis), discrete event system specification (DEVS) formalism and Petri Nets, with modelling and simulation of a system's behavior conducted using Petri Nets (Araz and Erden 2014).

Some authors map model-based practices with other concepts and techniques from simulation and control engineering. Isermann discusses a model-based practice, as the use of mathematical models helps to avoid manual tuning by trial and error (Isermann 1996a, b). Hence, from our understanding, his work supports model-based practices with system modelling techniques such as mathematical models. Isermann also mentions the use of Hardware-in-the-loop (HIL) simulation for both mechanical and electronic parts (Isermann 1996a, b). Later, in his 2006 publication, Isermann describes the use of a detailed V-model and includes the use of Software-in-the-loop, HIL and Rapid control prototyping (RCP), as well as object-oriented modelling (Isermann 2006). With some similarities, Lennon defends the use of model-based design to support mechatronic product development. Lennon developed a system-level model with block diagrams, a model that can be improved using data-driven empirical modelling (Lennon 2008). He also suggests the use of testing involving rapid control prototyping—called "rapid prototyping" in the article—and HIL simulation (Lennon 2008). RCP is also used by Chen et al. (2009) and Lapusan et al. (2010), and can be paired with HIL. Chen et al. (2009) also implement the design for control (DFC) method, which develops mechanical and controller design concurrently, as opposed to the design then control method (Chen et al. 2009). DFC was introduced and applied to mechatronics by Li et al. (2001). In a distributed control system context, Mahalik et al. discuss the use of "components-based design"—which, despite common principles, differs from the one used for CPS development in the next section. Components-based design mainly focuses on control design and on supporting RCP (Mahalik et al. 2006). Renner et al. present a design flow for mechatronic systems, with a focus on the information processing element. The design flow mainly composes with modelling and simulation through HIL simulation (Renner et al. 2000). The HIL advantages are also discussed in Veitl et al. (2000).

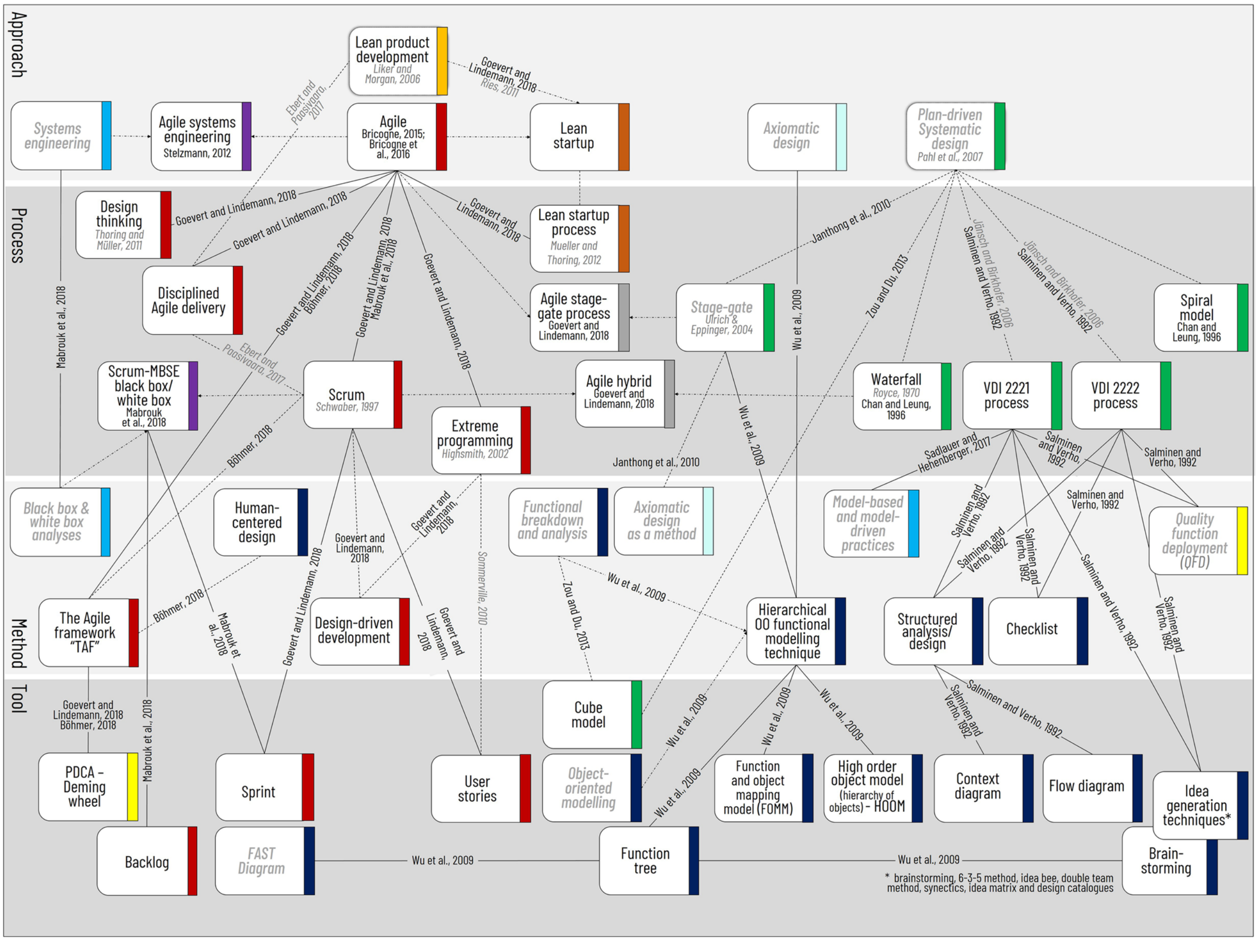
Approach
Process
Method
Tool
Lean product development Liker and Morgan, 2006
Goevert and Lindemann, 2018 Ries, 2011
Ebert and Paasivaara, 2017
Systems engineering
Agile systems engineering Stelzmann, 2012
Agile Bricogne, 2015; Bricogne et al., 2016
Lean startup
Axiomatic design
Plan-driven Systematic design Pahl et al., 2007
Design thinking Thoring and Müller, 2011
Goevert and Lindemann, 2018
Lean startup process Mueller and Thoring, 2012
Disciplined Agile delivery
Agile stage-gate process Goevert and Lindemann, 2018
Stage-gate Ulrich & Eppinger, 2004
Janthong et al., 2010
Zou and Du, 2013
Jänsch and Birkhofer, 2006
Salminen and Verho, 1992
Spiral model Chan and Leung, 1996
Mabrouk et al., 2018
Wu et al., 2009
Scrum-MBSE black box/ white box Mabrouk et al., 2018
Scrum Schwaber, 1997
Extreme programming Highsmith, 2002
Agile hybrid Goevert and Lindemann, 2018
Waterfall Royce, 1970 Chan and Leung, 1996
VDI 2221 process
VDI 2222 process
Böhmer, 2018
Sadlauer and Hehenberger, 2017
Black box & white box analyses
Human-centered design
Functional breakdown and analysis
Axiomatic design as a method
Model-based and model-driven practices
Quality function deployment (QFD)
Sommerville, 2010
The Agile framework "TAF"
Design-driven development
Hierarchical OO functional modelling technique
Structured analysis/ design
Checklist
Cube model
Goevert and Lindemann, 2018 Böhmer, 2018
PDCA - Deming wheel
Sprint
User stories
Object-oriented modelling
Function and object mapping model (FOMM)
High order object model (hierarchy of objects) - HOOM
Context diagram
Flow diagram
Idea generation techniques*
Backlog
FAST Diagram
Function tree
Brain-storming
Wu et al., 2009
* brainstorming, 6-3-5 method, idea bee, double team method, synectics, idea matrix and design catalogues

Fig. 7 Cartography of approaches, processes, methods and tools available for mechatronic product development—Focus on the Agile and Plan-driven—Systematic design approaches

Concepts and techniques related to modelling intersect with many other concepts and techniques. These are represented accordingly across Figs. 6, 7, 8, 9.

3.1.5 Object-oriented modelling and related concepts and techniques

Within modelling practices, object-oriented modelling can assist system engineers to create a system model by gathering components (Hamza et al. 2018). This model can be used to run simulations. Schramm et al. mainly focus object-oriented modelling and programming to support the V-model. Barbieri et al. consider object-oriented modelling, together with aspect-oriented modelling as "fundamental tools" for the design of mechatronics systems and the modelling of non-functional behaviors (Barbieri et al. 2016). They also discuss Software-in-the-loop and HIL simulation. Object-oriented design techniques are used by Counsell et al. (2002) to structure design knowledge, and then this knowledge is reused for complex system development as mechatronics during the conceptual design step. Accordingly, Counsell et al. use knowledge based-engineering within a stage-gate like systematic process proposed by French (1985).

Wu et al. also make use of object-oriented modelling within their framework, which they claim can be implemented within the six-step stage-gate process proposed by Ulrich and Eppinger (2004). Their framework relies on the proposed hierarchical OO functional modelling technique that integrates object-oriented and functional decompositions to support the concept development and system-level design steps of the process (Wu et al. 2009). The hierarchical OO functional modelling technique is considered as a method, as it is described as a procedure and takes place within the design process. Wu et al. also position their work within Axiomatic design. Accordingly, the function model supports the Axiomatic design's functional domain, the object model deals with the Axiomatic design's physical domain, and an information flow model is used to map the relation between the two domains. Hence, they suggest that the function model is supported by a function tree, and

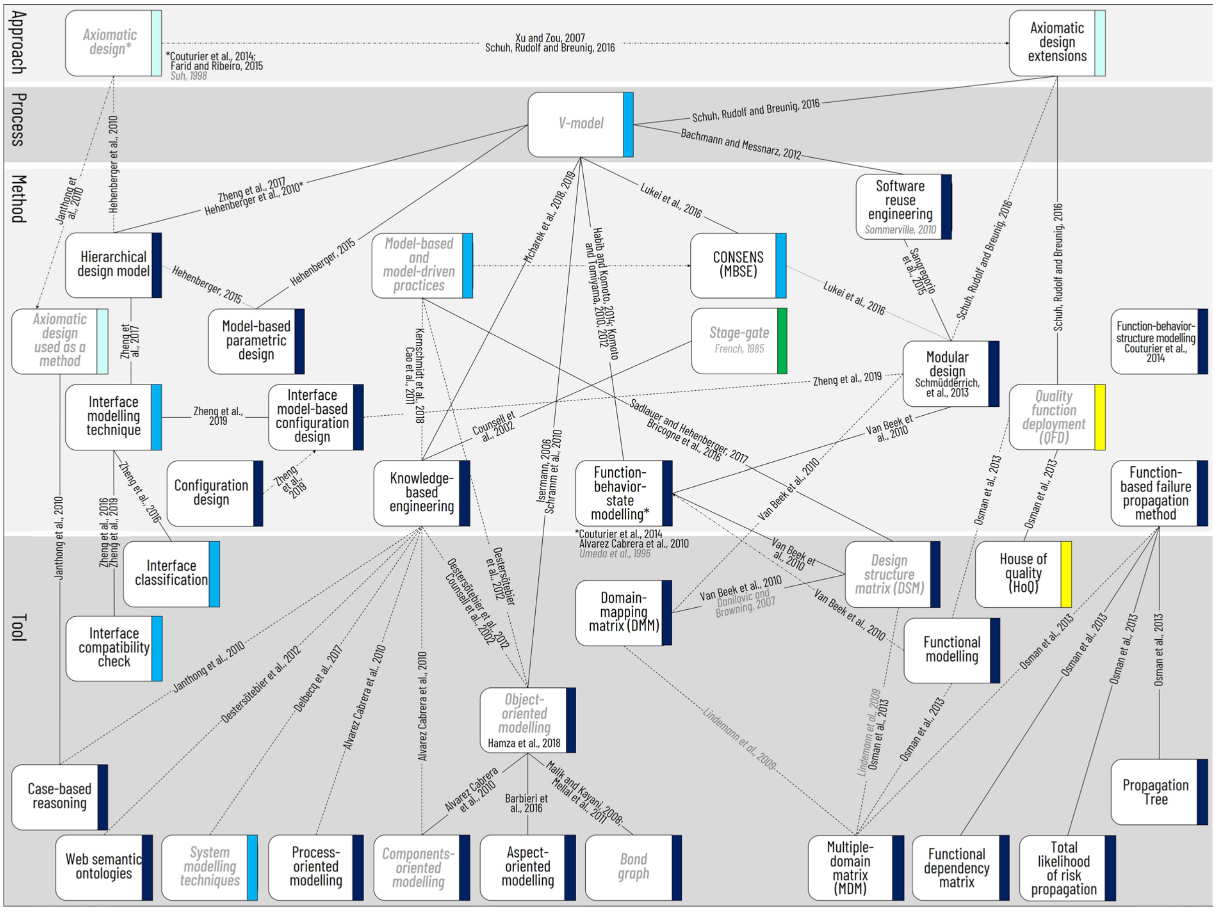
Approach
Process
Method
Tool
Axiomatic design*
*Couturier et al., 2014; Farid and Ribeiro, 2015
Suh, 1998
Xu and Zou, 2007
Schuh, Rudolf and Breunig, 2016
Axiomatic design extensions
V-model
Schuh, Rudolf and Breunig, 2016
Bachmann and Messnarz, 2012
Software reuse engineering
Sommerville, 2010
Janthong et al., 2010
Hehenberger et al., 2010
Zheng et al., 2017
Hehenberger et al., 2010*
Hierarchical design model
Hehenberger, 2015
Hehenberger, 2015
Model-based and model-driven practices
Mcharek et al., 2018, 2019
Habib and Komoto, 2014; Komoto and Tomiyama, 2010, 2012
Lukei et al., 2016
CONSENS (MBSE)
Lukei et al., 2016
Sangregorio et al., 2015
Schuh, Rudolf and Breunig, 2016
Schuh, Rudolf and Breunig, 2016
Axiomatic design used as a method
Zheng et al., 2017
Model-based parametric design
Kernschmidt et al., 2018
Cao et al., 2011
Stage-gate
French, 1985
Modular design
Schmüdderrich, et al., 2013
Function-behavior-structure modelling
Couturier et al., 2014
Zheng et al., 2019
Interface modelling technique
Zheng et al., 2019
Interface model-based configuration design
Counsell et al., 2002
Sadlauer and Hehenberger, 2017
Bricogne et al., 2016
Van Beek et al., 2010
Quality function deployment (QFD)
Zheng et al., 2016
Configuration design
Zheng et al., 2019
Knowledge-based engineering
Isermann, 2006
Schramm et al., 2010
Function-behavior-state modelling*
*Couturier et al., 2014
Alvarez Cabrera et al., 2010
Umeda et al., 1996
Van Beek et al., 2010
Osman et al., 2013
Osman et al., 2013
Function-based failure propagation method
Janthong et al., 2010
Zheng et al., 2016
Zheng et al., 2019
Interface classification
Oestersötebier et al., 2012
Oestersötebier et al., 2012
Counsell et al., 2002
Van Beek et al., 2010
Design structure matrix (DSM)
House of quality (HoQ)
Domain-mapping matrix (DMM)
Van Beek et al., 2010
Danilovic and Browning, 2007
Van Beek et al., 2010
Interface compatibility check
Functional modelling
Janthong et al., 2010
Oestersötebier et al., 2012
Delbecq et al., 2017
Alvarez Cabrera et al., 2010
Alvarez Cabrera et al., 2010
Object-oriented modelling
Hamza et al., 2018
Lindemann et al., 2009
Lindemann et al., 2009
Osman et al., 2013
Osman et al., 2013
Osman et al., 2013
Osman et al., 2013
Osman et al., 2013
Osman et al., 2013
Case-based reasoning
Alvarez Cabrera et al., 2010
Barbieri et al., 2016
Malik and Kayani, 2008; Mellal et al., 2011
Propagation Tree
Web semantic ontologies
System modelling techniques
Process-oriented modelling
Components-oriented modelling
Aspect-oriented modelling
Bond graph
Multiple-domain matrix (MDM)
Functional dependency matrix
Total likelihood of risk propagation

Fig. 8 Cartography of approaches, processes, methods and tools available for mechatronic product development—Miscellaneous concepts and techniques (1 of 2)

that the object model is represented with the help of the high-order object model (HOOM)—which allows a hierarchy of objects to be performed. The mapping is realized with the function and object mapping model (FOMM). The function tree can in turn be supported by a FAST diagram or by brainstorming (Wu et al. 2009); the latter is part of the "idea generation techniques" block and is highlighted by the overlap. This mapping is represented in Fig. 7.

Also linked with object-oriented modelling, bond graphs represent a possible tool for the modelling and simulation of systems (Malik and Kayani 2008; Mellal et al. 2011). The purposes for using this representation are varied. A bond graph can be used to represent the dynamic behavior of multi-domain systems, to support analysis and for simplification purposes, as in (Dauphin-Tanguy 2008; Mellal et al. 2011; Wu et al. 2012). Some authors combine bond graphs with other modelling techniques and languages such as block diagrams, activity diagrams or SysML (Schweiger et al. 1999; Turki et al. 2005). Schweiger et al. (1999) also apply state charts for information processing components, thereby combining bond graphs, block diagrams and state charts for the modelling of mechatronic systems. Turki et al. propose to integrate bond graphs as a SysML profile relying on activity and block diagram extensions.

Meanwhile, other authors combine bond graphs with genetic algorithms to conduct design space exploration to assess and select the optimal design architectures and configurations, or to enable design automation (Dupuis et al. 2015; Fan et al. 2004; Malik and Kayani 2008; Wang et al., 2005). In addition, for design space exploration, Gamage et al. propose to use linear graphs for system modelling (Gamage et al. 2011).

These concepts and techniques are gathered and displayed inn Figs. 6, 7, and 8.

3.1.6 Modular design and related concepts and techniques

In modular design, modularization can be based on the functional model of a system, itself performed using Function-Behavior-State modelling (Van Beek et al. 2010). According to those researchers, "[Function-Behavior-State] is a

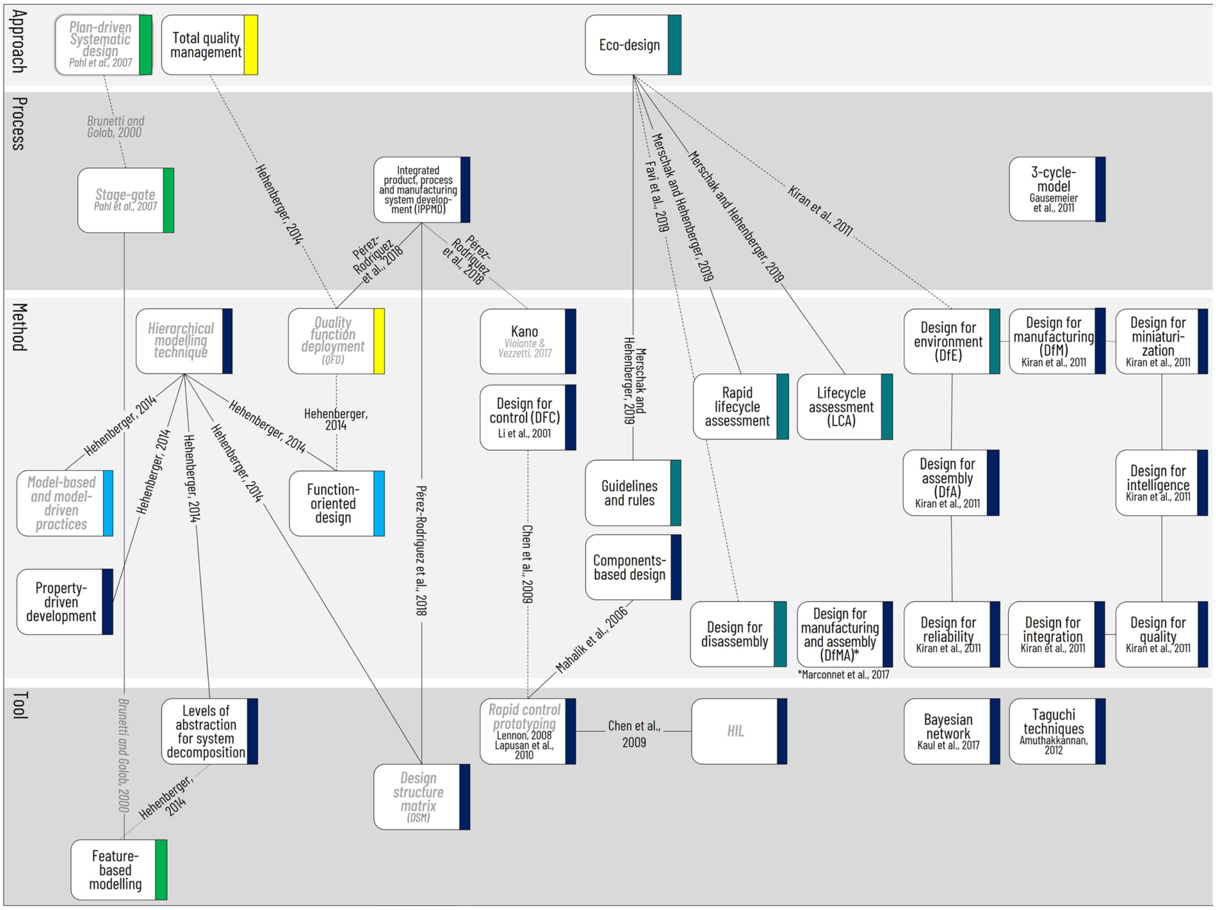
Approach
Process
Method
Tool
Plan-driven Systematic design Pahl et al., 2007
Total quality management
Eco-design
Brunetti and Golob, 2000
Hehenberger, 2014
Stage-gate Pahl et al., 2007
Integrated product, process and manufacturing system development (IPPMD)
Pérez-Rodríguez et al., 2018
Pérez-Rodríguez et al., 2018
Merschak and Hehenberger, 2019
Favi et al., 2019
Merschak and Hehenberger, 2019
Kiran et al., 2011
3-cycle-model Gausemeier et al., 2011
Hierarchical modelling technique
Quality function deployment (QFD)
Kano Violante & Vezzetti, 2017
Merschak and Hehenberger, 2019
Design for environment (DfE)
Design for manufacturing (DfM) Kiran et al., 2011
Design for miniaturization Kiran et al., 2011
Hehenberger, 2014
Design for control (DFC) Li et al., 2001
Rapid lifecycle assessment
Lifecycle assessment (LCA)
Hehenberger, 2014
Hehenberger, 2014
Hehenberger, 2014
Hehenberger, 2014
Hehenberger, 2014
Model-based and model-driven practices
Function-oriented design
Guidelines and rules
Design for assembly (DfA) Kiran et al., 2011
Design for intelligence Kiran et al., 2011
Pérez-Rodríguez et al., 2018
Chen et al., 2009
Property-driven development
Components-based design
Mahalik et al., 2006
Design for disassembly
Design for manufacturing and assembly (DfMA)*
*Marconnet et al., 2017
Design for reliability Kiran et al., 2011
Design for integration Kiran et al., 2011
Design for quality Kiran et al., 2011
Brunetti and Golob, 2000
Levels of abstraction for system decomposition
Rapid control prototyping Lennon, 2008 Lapusan et al., 2010
Chen et al., 2009
HIL
Bayesian network Kaul et al., 2017
Taguchi techniques Amuthakkannan, 2012
Hehenberger, 2014
Design structure matrix (DSM)
Feature-based modelling

Fig. 9 Cartography of approaches, processes, methods and tools available for mechatronic product development—Miscellaneous concepts and techniques (2 of 2)

particular type of function modeling" and, accordingly, Function-Behavior-State appears as a derivate of functional modelling. The Function-Behavior-State model is used by these authors to facilitate the construction of the DSM. In addition, they use the domain mapping matrix (DMM) (Danilovic and Browning 2007), a transformation matrix, to map the components with the modules (Van Beek et al. 2010). To enhance the cartography, Lindemann et al. (2009) map DSM and DMM with the multiple-domain matrix (MDM) also used by Osman et al. (2013). Osman et al. propose a framework composed of different methods and tools, such as quality function deployment (QFD) paired with the House of Quality (HoQ), functional modelling, and MDM. A DSM is also cited and linked with the MDM (Osman et al. 2013). Finally, for robust design purposes, Osman et al. deploy a function-based failure propagation method relying on tools such as a functional dependency matrix, propagation tree, the total likelihood of propagation, and from our understanding, MDM (Osman et al. 2013).

Finally, Sangregorio et al. consider that SE may not be suitable for small and customizable systems, and accordingly propose modular design and the reuse of software modules (Sangregorio et al. 2015).

The concepts and techniques discussed in this subsection are represented in Fig. 8.

3.1.7 Agile product development and related concepts and techniques

Agile has been studied by a number of authors with the goal of supporting mechatronic product development. In that sense, Bricogne investigates the use of Agile methods through a proposed Agile framework to "improve cooperative work and communication" (Bricogne 2015; Bricogne et al. 2016). This direction was followed by Goevert and Lindemann (2018), who state that the Agile values and principles are transferrable from software to mechatronic product development. Accordingly, they propose an Agile technique toolbox that encompasses "nine Agile processes and the integrated Agile techniques", including methods and tools, among which the authors list Scrum, Extreme programming (XP), Design thinking, Lean startup, "The Agile

Framework" (TAF), Agile hybrid model, Agile Stage-gate process, Disciplined Agile delivery (DAD) and Design-driven development.

Scrum, initially introduced by Schwaber (1997), can be envisioned as a process, as it aims at logically organizing a product's development. Scrum is a series of iterative steps with inputs (the backlogs) and outputs, with the achievement of an increment through the completion of the sprint backlog. The sum of each of the increments results in a product. Hence, Scrum is a process that can rely on user stories and sprints, both positioned as tools (Goevert and Lindemann 2018). Although Sommerville (2010) does not deal with the development of mechatronic products, XP can also be supported by user stories, which are short scenarios to identify requirements. XP can be considered as a process or as a method. Indeed, according to Highsmith, XP gathers 12 practices, which would make it closer to a method. Nonetheless, Highsmith (2002) also states that the goal of XP is to deliver high-quality software. Moreover, a workflow-like representation of XP exists, and from a description made by Sommerville (2010), XP is considered here as a process.

Design thinking has been presented by researchers as a series of steps leading to a tested prototype with a strong customer focus and possible iterations (Goevert and Lindemann 2018). Supported by the vision of Thoring and Müller (2011), Design thinking can be represented as a process model. Hence, Design thinking is considered as a process in this case, but other interpretations do exist.

Lean startup can be interpreted as both a process and as a hybrid approach of Agile and Lean. Accordingly, Lean product development is added to enhance the cartography (Liker and Morgan 2006). Ries claims that Lean startup is at the cross-roads of "lean manufacturing, design thinking, customer development, and agile development" and that it is a "new approach to creating continuous innovation" (Ries 2011). Ries also describes Lean startup as "the application of lean thinking to the process of innovation" and presents five principles. This statement would make Lean startup an approach. However, the interpretation from Mueller and Thoring presents Lean startup as a series of steps represented through a circlular process model (Mueller and Thoring 2012). Goevert and Lindemann also mention the six-step process. In this specific case, both representations of Lean startup are integrated (Goevert and Lindemann 2018).

TAF was introduced by Böhmer (2018) for mechatronic development. It is an agile prototyping framework that relies on existing concept and techniques. TAF aims at reducing both uncertainty and required effort through hypothesis testing and the realisation of prototypes. TAF is influenced by human-centered design (HCD) through the integration of three cycles: desirability, feasability, and viability. Each cycle is driven by the Plan-Do-Check-Act cycle—the Deming wheel, also known as the PDCA cycle—introducing the continuous improvement mindset (Goevert and Lindemann 2018). TAF can be incorporated into existing process models, called "sprints" by Böhmer (2018). Böhmer defines sprints as "short and time-boxed development cycles with limited scope". The sprint denomination is usually used when adressing the Scrum process. From our understanding, TAF can be classified as a method, as it is intended to be incorporated into an overall process, the author's "sprints". Accordingly, we understand sprints as Scrum's sprints, thus linking TAF with Scrum in the cartography.

Design-driven development lacks enough of a presence in the scientific literature to be accuratelly positioned. The description provided by Goevert and Lindemann would tend to position it on the method level, as the focus appears to be on requirement definition. Moreover, they introduce a link with both Scrum and XP (Goevert and Lindemann 2018).

DAD is a process, according to the description provided by Goevert and Lindemann (2018). To enhance the mapping of DAD, it is linked to both Scrum and Lean product development (Ebert and Paasivaara 2017).

At the crossroad of the works on Agile and SE, Stelzmann expounds on agility within SE for product integrating hardware systems and substantial software development. From our perspective, this can be interpreted as mechatronics. Stelzmann discusses Agile Systems Engineering and determines its application context (Stelzmann 2012). Similarly, Mabrouk et al. study the integration of Agile practices with the SE approach, especially the integration of the Scrum "method" within an "MBSE design methodology" (Mabrouk et al. 2018). From our understanding, the "MBSE design methodology" used is the two procedures of black box and white box analyses described by Mhenni et al.—see Sect. 3.1.1. Scrum is utilized to foster the iterative and collaborative aspects, and is inserted in between the black box and the white box analyses. The black box analysis builds up the product backlog, the latter builds the sprint's backlog. The sprint supports the white box analysis that leads to a physical architecture (Mabrouk et al. 2018). Moreover, the authors add a prototyping and a verification and validation step, making this Scrum-MBSE hybridization closer to a process, as it supports almost the entire product development (Mabrouk et al. 2018). These different concepts and techniques related to Agile are represented in Fig. 7.

3.1.8 Systematic design and related concepts and techniques

Before the expansion of SE and Agile approaches, the systematic design approach (Pahl et al. 2007), also referred to as plan-driven, was discussed by Salminen and Verho through the use of VDI 2221, which is a systematic stage-gate like process, and VDI 2222, which is a Pahl and Beitz stage-gate process model. However, the link with the systematic design approach is established from our understanding. In addition, these authors rely on methods such as the QFD, checklists, Structured analysis/design (SA/SD)—supported by context and flow diagrams –, and promote the use of idea generation techniques. Such techniques include brainstorming, the 6–3-5 method, idea bee, the double team method, synectics, idea matrix and design catalogues (Salminen and Verho 1992). For further reading on VDI 2221 and 2222, please see Jänsch and Birkhofer (2006). Presented as referring to the systematic approach, Zou and Du introduce a new functional representation by which their cube model is defined. Their cube model can be envisioned as an improvement of the systematic model from Pahl and Beitz to support functional reasoning during conceptual design (Zou and Du 2013).

While it is a part of the systematic approach, the Spiral model is also proposed to support mechatronics design (Chan and Leung 1996). Waterfall is mentioned as well, envisioned as an enhanced spiral model that incorporates concurrent engineering. Within the cartography, Waterfall is supported by Royce (1970).

These concepts and techniques are indicated in Fig. 7.

3.1.9 Axiomatic design and related concepts and techniques

Axiomatic design was initially proposed by Suh (1998). Xu and Zou rely on axiomatic design and propose an extension with time-domain and structure parameters for mechatronic systems (Xu and Zou 2007). Farid and Ribeiro propose to rely on axiomatic design for multi-agent reconfigurable mechatronic systems (Farid and Ribeiro 2015). Janthong et al. propose to implement Axiomatic design within a systematic design process. The systematic design is used to structure the process in several stages—interpreted as a Stage-gate like process; whereas axiomatic design supports the mapping between functional and structural features during process' stages. Hence, in this particular case, axiomatic design is used as a method. Axiomatic design also enables the knowledge structure to store and compare cases. Indeed, the functional and physical structure of past projects are stored in a case library. Product architectures aim to be reused in future conceptual design stages. Janthong et al. also combine case-based reasoning and knowledge reuse (Janthong et al. 2010). Axiomatic design was extended by Schuh et al. (2016a, b) for the design of modular product platforms in mechatronic systems. From our understanding, modular design is performed through an axiomatic design extension, including the mechatronic function module. Schuh et al. (2016a, b) also mention the V-model, and the possible application of QFD to derive the functional requirements from the customer attributes. From an overall standpoint on axiomatic design, as only product development is discussed, most of the above-mentioned authors only address the functional and physical domains (Schuh et al., 2016a, b).

Also related to the axiomatic design approach, Hehenberger et al. propose a hierarchical design model to support the decomposition and mapping of design parameters and functional requirements (Hehenberger et al. 2010). This work was developed specifically for mechatronic design purposes, and the mechatronic design process cited by Hehenberger et al. (2010) was apparently the V-model. Hehenberger (2014) later introduced different methods that can be combined with hierarchical modelling, such as Function-oriented design, which can be combined with Total Quality management methods such as QFD, as well as model-based design and property-driven development, noted as “property-driven development/design” by Hehenberger. The levels of abstraction for system decomposition and DSM in regards to the tools are also presented to support the proposed hierarchical modelling technique. Linked with levels of abstraction, Hehenberger cites feature-based modelling, which is considered here as a tool (Hehenberger 2014). Utilizing backwards snowballing on Hehenberger's 2014 article, feature-based modelling can be linked to the stage-gate process and the systematic design approach from Pahl and Beitz (Brunetti and Golob 2000; Pahl et al. 2007). However, Brunetti and Golob do not address mechatronic product development.

Finally, to summarize axiomatic design for mechatronics, Alvarez Cabrera et al. question its use with their statement that “mechatronic products implement an increasing number of functionalities while maintaining constraints on space and costs, and thus, a tight integration of the subsystems is desirable, which makes it harder to obtain functional independence” (Alvarez Cabrera et al. 2010).

These different concepts and techniques are illustrated in Figs. 7, 8 and 9.

3.1.10 Eco-design and related concepts and techniques

More recently, the Eco-design approach has gained increasing interest within mechatronics design with the work of Merschak and Hehenberger. In their 2019 article, they discuss the integration of Eco-design practices such as guidelines and rules, rapid lifecycle assessment, detailed lifecycle assessment (LCA), and the different ways of supporting and implementing them (Merschak and Hehenberger 2019). In a work likely related to the Eco-design approach, Favi et al. (2019) propose to make use of Design for Disassembly (DfD) for mechatronic products applied through rules and guidelines. Design for X guidelines are suggested by some authors to support mechatronic product development, while Marconnet et al. (2017) propose using Design for Manufacturing and Assembly (DfMA). Kiran et al. (2011)

deploy in a concurrent engineering manner eight Design for X "abilities": Design for Integration, Design for Miniaturization, Design for Manufacturing (DfM), Design for Assembly (DfA), Design for Intelligence, Design for Environment (DfE), Design for Reliability, and Design for Quality. Each ability is represented with design parameters, which, in some cases, are specified for mechanical, electrical/electronic or software engineering. These eight Design for X abilities are represented in a cycle within the cartography.

All of the above-mentioned concepts and techniques are represented in Fig. 9.

3.1.11 Integrated product development

Various authors have promoted integrating different considerations into product development. Pérez-Rodríguez et al. (2018) propose the use of an Integrated Product, Process and Manufacturing System Development (IPPMD) reference model combined with DSM, Kano, and QFD for "analysis of the functional requirements". While the authors do not provide enough information regarding the use of Kano, they do mention a reference that supports considering Kano as a method, supported by tools such as the Kano questionnaire, Kano evaluation table, and the Kano category result (Violante and Vezzetti 2017). The IPPMD reference model is considered here as a process, as it serves to organize the whole product development concurrently with process and manufacturing developments. In a similar direction, Gausemeier et al. (2011) propose an integrative development illustrated with the 3-Cycle-Model designed to manage the whole product development "on the highest level of abstraction" and hence is considered as a process. This integrated process encompasses business, product and production developments as well as their different relations and synchronizations. Also linked with the integration of product and production developments, Lukei et al. propose to concurrently design product and production equipment based on concurrent, linked V-models, combined with the Conceptual Design Specification Technique for the Engineering of Complex Systems (CONSENS), a MBSE derivate. According to Lukei et al., designed products are intended to be modular and MBSE is used to support this modularization (Lukei, Hassan, Dumitrescu, Sigges & Derksen, 2016). These concepts and techniques are represented in Fig. 8 and in Fig. 9.

3.1.12 Other concepts and techniques

For the early integration of safety as well as risk assessment in the design process, Sierla et al. propose the Functional Failure Identification and Propagation framework. This framework relies on a functional system model, a configuration model and a behavior model, and is proposed to complement Fault Tree Analysis, FMEA and Probabilistic risk assessment (Sierla et al. 2012). This work is represented in Fig. 6. To deal with the reliability concerns of mechatronics systems, Kaul et al. (2017) propose an integrated model encompassing a behavior model—a subset of the system model—and a reliability model supported by a Bayesian network (Kaul et al. 2017). For optimal and robust design, the work of Amuthakkannan (2012) relies on the implementation of Taguchi techniques. These concepts and techniques are represented in Fig. 9.

Section 3.1 surveyed the variety of concepts and techniques proposed in the scientific literature to support mechatronic product development. Based on the concepts and techniques presented here, a graphical representation was established and is presented in detail in the next section.

3.2 Cartography of concepts and techniques for mechatronic product development

The previous sub-sections presented the variety of concepts and techniques available for mechatronic product development according to the scientific literature. This section organizes them, as well as their relations with each other, within a cartography based on the four-level model. Some concepts and techniques, such as Lean product development (Liker and Morgan 2006), as well as some references, are added to enhance the overall comprehension and structuration. For complementary information regarding these blocks and references, please refer to Sect. 2.3 and Fig. 4. Moreover, given that mechatronic product development covers several concepts and techniques, it is represented in different figures to support some clarity and to facilitate knowledge transfer. Accordingly, Figs. 6, 7, 8, 9 represent the cartography of concepts and techniques for mechatronic product development. Figure 6 represents the SE *set* and the concepts and techniques that it can include. Figure 7 mainly focuses on Agile and plan-driven—systematic design, and Figs. 8 and 9 present miscellaneous concepts and techniques and isolated blocks.

A first observation is that mechatronic product development encompasses a large variety of approaches, including Eco-design, Total quality management, SE, Axiomatic design, Agile and some of their hybridizations such as Agile systems engineering and Lean startup (Figs. 6,7,8). While mechatronic product development offers different hybridizations at the approach and process levels, identified with the grey, orange, and purple rectangles, it only aggregates a single *set*, i.e., only one approach is supported by a combination of concepts and tools connecting at least a process, a method and a tool. This *set* is the SE *set*, comprised of the V-model as a process, and model-based and model-driven practices as methods, which in turn rely on numerous modelling techniques used as tools, especially the "system modelling techniques" block (see Figs. 6 and 8). This aggregation can be envisioned as the SE *set's backbone*. As illustrated in Fig. 6, the SE *set's backbone* is supported by numerous studies, which suggests an emerging consensus around it, and thereby allows a possible recommendation for its use for mechatronic product development.

As an integral part of the SE *set*, models are an acknowledged means with which to manage the ever-increasing product complexity. The principles implemented within the SE approach and combined with the V-model can also be used to deal with product complexity. In addition to the SE *set's backbone*, DSM can also be used to manage product complexity; some authors attach DSM to model-based and model-driven practices (see Fig. 8). As shown in Fig. 8, different authors explore a range of modelling techniques, including process-oriented, aspect-oriented, object-oriented, and functional modelling. Modelling languages such as UML and SysML are discussed as well, and are considered as part of the "system modelling techniques" block, as presented in Fig. 6 and in Fig. 8. Indeed, before the apparition/democratization of SysML and UML, some authors (e.g., Buur and Andreasen 1989) had already defended the introduction of models. Hence, the "system modelling techniques" block represents the authors who acknowledge modelling as part of mechatronic product development or discuss modelling languages. This block is considered as a tool and can support the deployment of model-based and model-driven practices (Fig. 6). However, some authors explicitly mention the use of block diagrams for architecture and the use of state charts or activity diagrams for dynamic modelling, as they offer details about the type of diagrams and modelling used, i.e., a subset of SysML. These tools are therefore represented separately, as they specify some diagrams and specific aspects of modelling techniques. Petri nets and bond graphs are also represented as separate items. Indeed, the modelling languages listed above are focused on semantic aspects, whereas, despite containing a semantic part, Petri nets can be considered as focused on dynamic modelling (Habchi and Barthod 2016) and bond graphs are used to represent components and multi-physics flows (Wu et al. 2012).

Beyond the modelling techniques and components of the proposed SE *set* which have attracted significant research interest, practices with respect to the notion of function are also discussed by various authors. These include functional breakdown and analysis (Fig. 6 and Fig. 7), functional modelling (Fig. 8), function-oriented design and (Fig. 9), Function-Behavior-State modelling and Function-Behavior-Structure modelling (Fig. 8), as well as QFD (Figs. 7, 8 and 9) and the work proposed by Wu et al (2009) in Fig. 7. Moreover, part of these agnostic concepts and techniques—dark blue rectangle—can be attached to and enrich the SE *set's backbone*. For instance, functional analysis and breakdown and Function-Behavior-State modelling can support the V-model. Similarly, as depicted in Fig. 8, knowledge-based engineering is also discussed by several authors and is linked to different modelling practices. Indeed, stored knowledge extracted from previous models can in turn allow models to be built faster and with less errors through re-instantiation. Some other concepts and techniques should also be mentioned. HIL and modular design are reviewed by different authors and could find their place within the SE *set* as well, via links with the V-model for HIL (Fig. 6), and through CONSENS, an MBSE derivate, for modular design (Fig. 8).

Mechatronic product development is also supported by three incomplete *sets*: Agile (Fig. 7), Axiomatic design (Figs. 7 and 8), and systematic design—also called the plan-driven approach (Figs. 7 and 9). Agile literature for mechatronic product development—identified with a red rectangle—is essentially composed of processes, with only a few methods and tools indicated to support the process operationalization. Agile lacks an integration between the different levels and so cannot be considered as a set. Axiomatic design approach complemented by the proposed extensions and derivations as methods represents the very few blocks that belong to the axiomatic design approach itself—the turquoise rectangle (see Figs. 7 and 8). However, axiomatic design can rely on a few concepts and techniques that support the navigation between domains or their breakdown (Schuh et al. 2016a, b; Wu et al. 2009). The systematic design approach (Pahl et al. 2007), identified with a green rectangle, is not formally named, but processes, methods and tools are mapped and prefigure a possible *set*. Part of this incomplete *set*, Salminen and Verho (1992) are among the first authors to gather and structure a coherent group of processes, methods, and tools, possibly integrating the approach level for mechatronic product development (Fig. 7). In the cartography, Goevert and Lindemann follow a similar path with a structure of Agile concepts and techniques, named a "toolbox" for mechatronic design (Fig. 7). These works and their implementation are a partial answer to our research questions and lend credence to the need to support companies moving towards MPD.

Among the other approaches, Eco-design, identified with a dark green rectangle and represented in Fig. 9, is only discussed via some methods. The presence of the total quality management approach can also be observed. However, its concepts and techniques, identified with a yellow rectangle, such as QFD, the House of Quality, and PDCA are scattered and rarely mapped together—see Figs. 7, 8 and 9. For instance, QFD and PDCA are mainly attached to other concepts and techniques structures, especially incomplete *sets*. Precisely, QFD is attached to the systematic design and the axiomatic design incomplete *sets* (Figs. 7 and 8), whereas PDCA is linked to the Agile incomplete *set* (Fig. 7).

In addition, among the concepts and techniques proposed for mechatronic product development, it is possible to distinguish three different engineering backgrounds at the method and tool levels: mechanical engineering, control engineering, and software engineering. A mechanical engineering background is represented through methods and tools such as functional breakdown and analysis (Plateaux et al. 2010) supported by FAST and SADT, or APTE supported by Octopus diagrams and Horned beast, or through DfX practices (Fig. 9), which are also mentioned by some authors. A control engineering background is represented through Rapid control prototyping, HIL and Design for control. Finally, some concepts and techniques are rooted in software engineering, such as user stories, XP, and Scrum. This observation tends to confirm the statement made by Sadlauer and Hehenberger that "current publications on mechatronics reflect the past influence of mechanical engineering" and "with the rapid advances in information technologies, the significance of the methods in this field will steadily increase the use of design tools and methods from software engineering in mechatronics" (Sadlauer and Hehenberger 2017).

An evolution in the maturity of mechatronic product development can also be observed. Listing the challenges of mechatronic design, Alvarez Cabrera et al. (2010) mention a "lack of tools and methods supporting multi-disciplinary design". Based on the established cartography, the question of a lack of concepts and techniques to support MPD can be discussed. Indeed, mechatronic product development can be supported by a variety of concepts and techniques that are gathered within the cartography.

To synthesize, mechatronic product development is well-defined within the scientific literature, and can be supported by the SE *set*, whose *backbone*—composed of the V-model, model-based and model-driven practices, and system modelling techniques—is enriched by multiple concepts and techniques. Hence, the SE *set* is substantiated by a significant number of references, allowing a possible recommendation regarding its use to be envisioned. Finally, three incomplete *sets*: Agile, Plan-driven—systematic design and Axiomatic design, contribute conspicuously to the mapping.

The next section focuses on cyber-physical systems, another type of product benefitting from multidisciplinary development.

3.3 Concepts and techniques for cyber-physical systems' development

Before defining cyber-physical systems (CPS) and exploring their development concepts and techniques, it is worth noting that cybernetics was coined by Norbert Wiener in 1948 (Wiener 1965). Making the connection with industrial revolutions, this term appeared at the dawn of the third industrial revolution of the 1950's (Bloem et al. 2014) (or that of around 1960 according to Bauernhansl (2014) and Drath and Horch, (2014)). Cybernetics was introduced to provide a name to the field of control and communication theory in both natural and artificial systems. Through the definitions of "analogy and numerical machines", cybernetics is apparently the source of numerous research domains, such as artificial intelligence, control engineering, and computer science (Wiener 1965; Bricogne 2015). The "cyber-" denomination recently regained popularity through the appearance of CPS. CPS are, compared to mechatronics and cybernetics, a type of multidisciplinary products introduced much more recently, by Helen Gill in 2006 at the National Science Foundation in the USA, and refers to the integration of computation with physical processes (Lee and Seshia 2011). CPS are envisioned as "a new generation of systems with integrated computational and physical capabilities that can interact with humans through many new modalities" (Baheti and Gill 2011). Other definitions have since emerged to specify and conceptualize CPS's abilities and capacities (Rajkumar 2012; Liu et al. 2017; Rúbio et al. 2019). CPS are expected to impact several areas, including transportation, healthcare, defense and aerospace (Darwish and Hassanien 2018). Common to the plethora of definitions in the literature, CPS integrate both software and hardware parts. It is acknowledged that software and hardware have different development processes and different lifecycles (Blanchard & Fabrycky, 1990; Crnkovic, Asklund & Dahlqvist, 2003). Based on a similar statement, Broy and Schmidt proposed that "the transition from physical products to cyber-physical systems requires a fundamentally new engineering approach" (Broy and Schmidt 2014). In that sense, this section presents different concepts and techniques proposed in the scientific literature for CPS development.

Section 3.3.1 offers a brief overview of human- and user-centered design and related concepts and techniques, introducing SE and W-model. Sections 3.3.2 and 3.3.3 complete the SE-related practices with the introduction of model-based and model-driven practices and modelling techniques, respectively. Section 3.3.4 discusses the Agile concepts and techniques. Section 3.3.5 presents the triptych platform-, component- and contract-based design, as well as adjacent concepts and techniques. Finally, Section 3.3.6 addresses miscellaneous concepts and techniques.

3.3.1 Human- and user-centered design and related concepts and techniques

To support companies willing to make the transition from physical products to CPS, Broy and Schmidt propose that companies reinvent "their innovation and development processes and take a user-centric engineering approach" (Broy and Schmidt 2014). A similar direction is encountered in the medical domain, which has a strong emphasis on the human and accordingly justify the use of methods such as

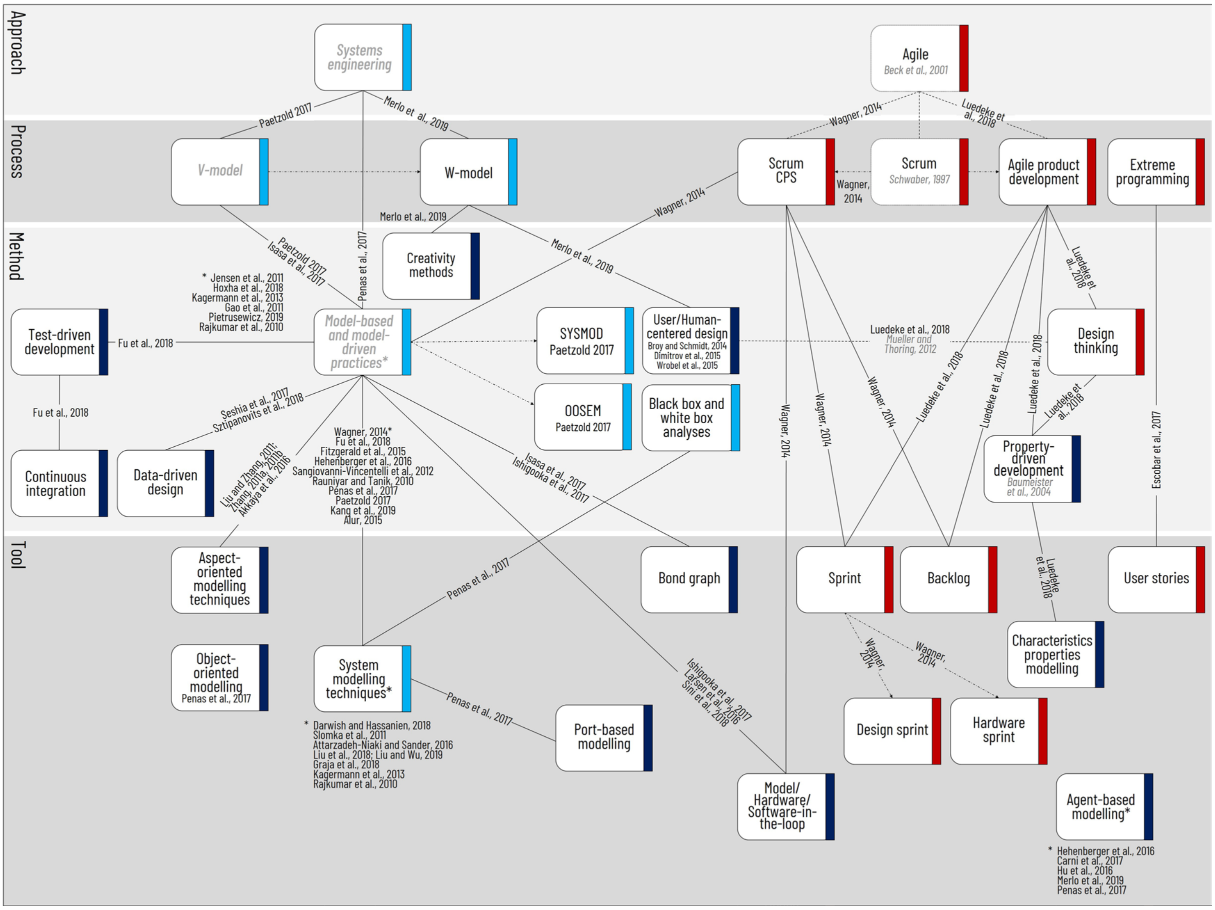

Fig. 10 Cartography of approaches, processes, methods and tools available for CPS development—focus on SE and Agile concepts and techniques

user-centered design (Wrobel et al. 2015) or HCD (Dimitrov et al. 2015). Merlo et al. propose to combine HCD criteria and creativity methods with an SE approach conducted through a "W-cycle" (Merlo et al. 2019); an approach that combines a process with and methods for CPS development. These proposals are represented in Fig. 10.

3.3.2 Model-based and model-driven practices and related concepts and techniques

Other authors support the idea that changes are required to develop these new products (Anderl, 2014; Fitzgerald et al. 2015; Song et al. 2019). Fitzgerald et al. (2015) formulate recommendations regarding CPS design process and engineering, focussing on an "integrated tool chain for model-based CPS" that integrates modelling, analysis, simulation and testing, as well as implementation. This "tool chain" is represented on the cartography (Fig. 10) as the joint usage of model-based practices—"model-based and model-driven practices" block—with system modelling techniques.

Model-based practices as a support for CPS development have been studied by different authors. Accordingly, Jensen et al. propose a ten-step model-based design for CPS including both software and hardware aspects, as well as verification, validation, tests and simulation aspects (Jensen et al. 2011). Rajkumar et al. (2010) discuss the use of model-based development, as well as the necessity to integrate verification and validation. The authors specify that not only software should be modelled, but also "communications, computing and physical dynamics", covering some new capabilities of CPS (Rajkumar et al. 2010). Similarly, Ishigooka et al. (2017) make use of model-based development for safety critical CPS, paired with HIL simulation—integrated in the "Model/Hardware/Software-in-the-loop" block—and bond graphs. In a similar direction, some authors also promote the use of model-based development and analysis supported by system modelling techniques (Kang et al. 2019; Sangiovanni-Vincentelli et al. 2012). According to Pietrusewicz (2019), model-based development is useful when "the modelled system is large, complex or cross-domain in nature". Paetzold (2017) discusses the usage of MBSE—represented within the "model-based and model-driven practices" block—for CPS development. Two MBSE derivations are also discussed: the OOSEM (Object-oriented Systems Engineering Method) and the "SYSMOD". Paetzold relies on system modelling techniques to support MBSE. Furthermore, MBSE is part of a broader perspective carried by the V-model, which in turn is integrated in the SE approach.

Using the possibilities offered by data collection, Seshia et al. propose to combine model-based design with data-driven design (Seshia et al. 2017). To this extent, model-based design can be enriched from collected data and facilitates the system's adaptation to an environment that is difficult to model. This integration is also discussed by Sztipanovits et al. (2018) as a research opportunity. Data-driven design as a support of model-based design can be viewed as a tool or as a method. From our understanding, Seshia et al. propose to combine data-driven design with model-based design, which can be considered as an engineering practice and thus would position data-driven design as a method. This is the classification adopted and represented within the cartography. Also related to the enhancement of model-based practices, Isasa et al. focus on the integration of energy-consumption concerns within model-based development. Bond graphs can be used for the modelling. These authors also position their work within a V-model (Isasa et al. 2017).

To ensure reliable and robust CPS and to address non-functional properties, Fu et al. (2018) envison a combination of model-based design relying on UML paired with test-driven development (TDD). They also mention continuous integration as a good practice (Fu, Choosilp & Dong, 2018). Another work proposes using model-based design combined with aspect-oriented modelling techniques (Akkaya et al. 2016). Aspect-oriented programming (integrated to aspect-oriented modelling in the cartography) is also used with model-driven architecture by (Liu & Zhang 2011a, b; Zhang, 2011a, 2011b). This combination aims at enhancing systems' modularity and helps to deal with complexity and non-functional requirements.

Sztipanovits et al. (2018) introduce the OpenMETA tool suite to support component-based and model-based design for CPS development. In addition, the authors defend the reuse of design knowledge—referred to as knowledge-based design in the cartography—available through component model libraries. In a similar direction, INTO-CPS is an integrated tool chain to support model-based design and analysis that also integrates Hardware and Software-in-the loop simulation (Larsen et al. 2016)—integrated in the "Model/Hardware/Software-in-the-loop" block.

Hence, model-based practices, such as model-based development and model-based design, are acknowledged methods to support CPS development (Gao et al. 2011; Hoxha et al. 2018; Kagermann et al. 2013; Pietrusewicz 2019). Model-based practices make it possible to cope with complexity and offer the early verification/validation of design (Seshia et al. 2017). Despite a common agreement within the scientific literature on the use of model-based practices, some authors take a contrary direction. Raghav and Gopalswamy state that "model-based development approaches are inadequate for complex CPS" and propose to utilize an Architecture-driven development method instead (Raghav and Gopalswamy 2009).

These different concepts and techniques are mostly represented in Fig. 10, except for the works of Sztipanovits et al. (2018) and Raghav and Gopalswamy (2009) works, represented in Fig. 11.

3.3.3 Modelling techniques

System modelling plays a crucial role in CPS development (Darwish and Hassanien 2018). Modelling allows developers to deal with products' increasing complexity (Kagermann et al. 2013; Alur 2015). As stated throughout the previous paragraphs, the implementation of model-based and model-driven practices generally requires the implementation of modelling techniques and languages. For instance, Alur discusses model-based design supported by system modelling techniques (Alur 2015). Similarly, Rauniyar and Tanik state that UML "provides the foundation of the Model-Driven Architecture" (Rauniyar and Tanik 2010). Slomka et al. focus on system modelling techniques to build up architectures through a new system description language introducing symbols of communication, interfaces and requirements (Slomka et al. 2011). This avenue for creating new modelling techniques—languages and tools –to "specify, analyze, synthesize and simulate different compositions" is envisioned by (Rajkumar et al. 2010).

To further investigate modelling practices, Hehenberger et al. study the development of CPS from a model and modelling perspective (Hehenberger et al. 2016). The authors mention the use of MBSE, which can be supported by SysML or Architecture Description Language (ADL). Within the ADL family, Liu et al. use the architecture analysis and design language (AADL) (Liu et al. 2018; Liu and Wu, 2019). Attarzadeh-Niaki and Sander propose the ForSyDe framework, relying on SystemC language to model and simulate complex systems (Attarzadeh-Niaki and Sander 2016), and Graja et al. propose a survey on CPS modelling techniques (Graja, Kallel, Guermouche, Cheikhrouhou & Hadj Kacem, 2018).

Penas et al. envision CPS design as a possible adaptation of mechatronics design to cope with their differences (Penas et al. 2017). They discuss adopting a "Systems Engineering viewpoint" to consider the whole lifecycle of a system. This

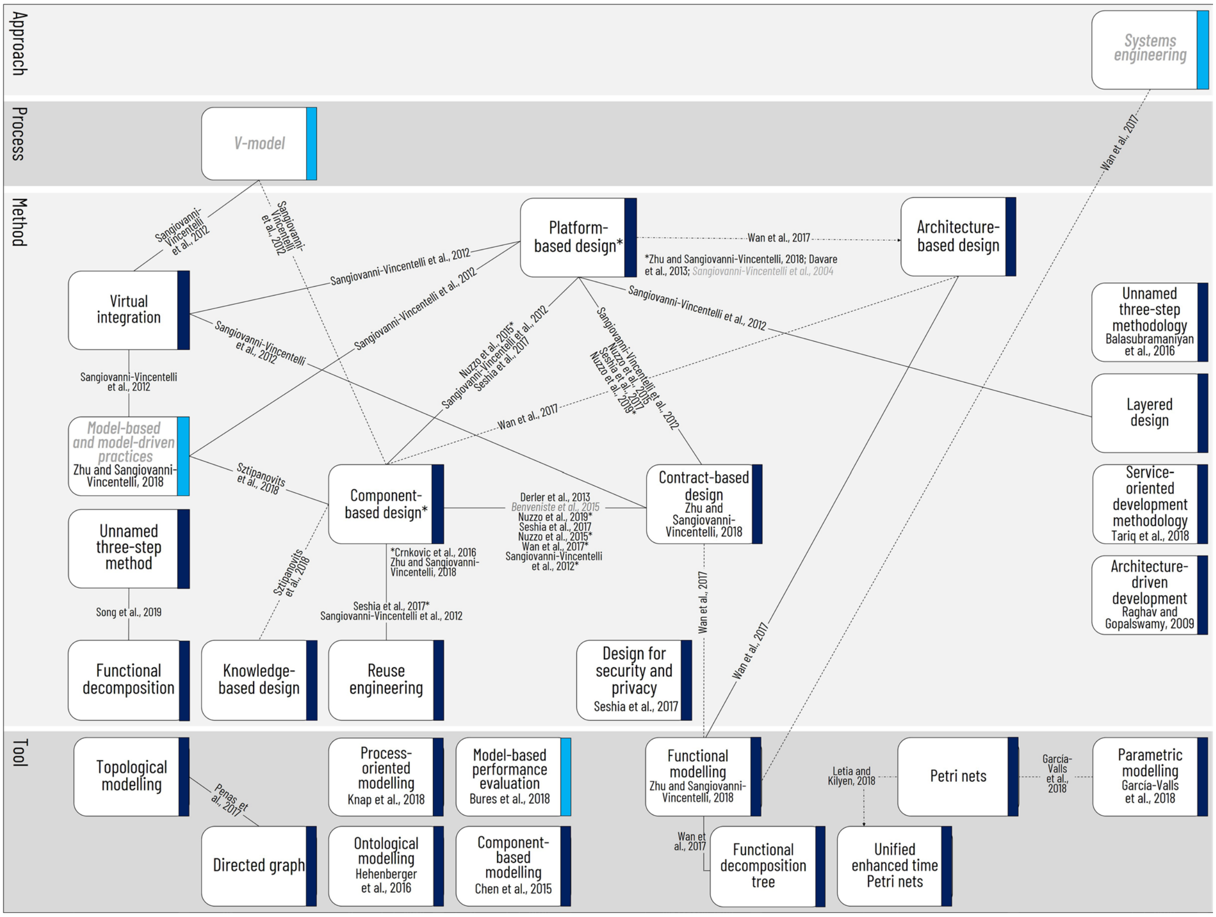
Approach
Process
Method
Tool
Systems engineering
V-model
Wan et al., 2017
Sangiovanni-Vincentelli et al., 2012
Sangiovanni-Vincentelli et al., 2012
Platform-based design*
Wan et al., 2017
Architecture-based design
*Zhu and Sangiovanni-Vincentelli, 2018; Davare et al., 2013; Sangiovanni-Vincentelli et al., 2004
Virtual integration
Sangiovanni-Vincentelli et al., 2012
Sangiovanni-Vincentelli et al., 2012
Sangiovanni-Vincentelli et al., 2012
Sangiovanni-Vincentelli et al., 2012
Nuzzo et al., 2015*
Sangiovanni-Vincentelli et al., 2012
Seshia et al., 2017
Sangiovanni-Vincentelli et al., 2012
Nuzzo et al., 2015
Seshia et al., 2017
Nuzzo et al., 2019*
Sangiovanni-Vincentelli et al., 2012
Unnamed three-step methodology
Balasubramaniyan et al., 2016
Sangiovanni-Vincentelli et al., 2012
Wan et al., 2017
Layered design
Model-based and model-driven practices
Zhu and Sangiovanni-Vincentelli, 2018
Sztipanovits et al., 2018
Component-based design*
Derler et al., 2013
Benveniste et al., 2015
Nuzzo et al., 2019*
Seshia et al., 2017
Nuzzo et al., 2015*
Wan et al., 2017*
Sangiovanni-Vincentelli et al., 2012*
Contract-based design
Zhu and Sangiovanni-Vincentelli, 2018
Service-oriented development methodology
Tariq et al., 2018
Unnamed three-step method
*Crnkovic et al., 2016
Zhu and Sangiovanni-Vincentelli, 2018
Sztipanovits et al., 2018
Architecture-driven development
Raghav and Gopalswamy, 2009
Song et al., 2019
Seshia et al., 2017*
Sangiovanni-Vincentelli et al., 2012
Wan et al., 2017
Wan et al., 2017
Functional decomposition
Knowledge-based design
Reuse engineering
Design for security and privacy
Seshia et al., 2017
Topological modelling
Penas et al., 2017
Process-oriented modelling
Knap et al., 2018
Model-based performance evaluation
Bures et al., 2018
Functional modelling
Zhu and Sangiovanni-Vincentelli, 2018
Letia and Kilyen, 2018
Petri nets
García-Valls et al., 2018
Parametric modelling
García-Valls et al., 2018
Directed graph
Ontological modelling
Hehenberger et al., 2016
Component-based modelling
Chen et al., 2015
Wan et al., 2017
Functional decomposition tree
Unified enhanced time Petri nets

Fig. 11 Cartography of approaches, processes, methods and tools available for CPS development—Miscellaneous concepts and techniques

could be supported by MBSE, which incorporates multi-view considerations. This multi-view aspect is supported by system modelling techniques conducted in SysML, thereby allowing the dynamic structures of CPS to be represented. System modelling techniques are also used to support black box and white box analyses. Object-oriented modelling is also mentioned as a means to create behavioral models using causal or acausal modelling. To model the interactions between the subsystems, port-based modelling is introduced and implemented within SysML diagrams. Topological modelling is also discussed to deal with complexity through the use of graph theory and especialy the directed graph—digraph—to represent CPS hierarchical structures. The authors develop a strong focus on modelling practices and how these practices can contribute to CPS development (Penas et al. 2017).

Also linked with modelling techniques, Knap et al. expand process-oriented modelling from organization management to CPS design (Knap et al. 2018). For self-adaptive CPS design, García-Valls et al. support on-line adaptability through the use of parametric models, whereas modelling and verification are based on Petri nets (García-Valls et al. 2018). Letia and Kilyen (2018) also rely on Petri nets combined with fuzzy logic and rule-based systems to propose Unified Enhanced Time Petri Nets models.

As underscored by Bures et al., modelling is general and encompasses different focuses. Modelling can be performed for functional analysis or verification purposes, which has been well-documented in the literature. However, the performance analysis purposes are neglected according to Bures et al., who propose a “model-based performance evaluation” of CPS (Bures et al. 2018). In addition, ontological modelling and agent-based orientations can also complement the list of modelling concepts and techniques (Hehenberger et al. 2016). Penas et al. also discuss the use of multi-agent modelling, as it can help CPS achieve “complexity management, flexibility, robustness, adaptation and reconfigurability” (Penas et al. 2017). The “agent orientation as a modelling and engineering paradigm is currently completely applicable” (Hehenberger et al. 2016) and is acknowledged for its efficiency in terms of complexity management. Based on the agent paradigm, Carni et al. (2017) propose and implement an architecture for CPS development. The use of the agent-based orientation for CPS development is also promoted by Hu et al. (2016) and envisioned by Merlo et al. (2019) to model and simulate CPS. To support the development of multi-agent systems for software architecture, Sini et al. propose relying on a combination of model-based software design techniques paired with Model/Software/Hardware-In-The-Loop, structured in an iterative workflow (Sini et al. 2018). The use of the agent-based orientation as a modelling practice or as an architecture is taking on a growing importance for CPS development. It is worth mentioning that the agent-based orientation, as with test-driven development and continuous integration methods mentioned above, has a software background. Also linked to software engineering practices, Tariq et al. (2018) make use of a Service-oriented development methodology.

The techniques presented in this subsection are represented in Figs. 10 and 11.

3.3.4 Agile concepts and techniques

As with mechatronics, software concepts and techniques are used for CPS development. The Agile approach is implied by some authors but not named. Within the cartography, and to enhance comprehension, Agile positioning is based on the Agile manifesto (Beck et al. 2001). At the process level, Scrum (Schwaber 1997) can be used (with an adaptation) to handle concurrent hardware and software design. This adaptation is labelled as Scrum CPS and is supported by model-based development and the use of models and simulation. Regarding the implemented tools, Wagner discusses the deployment of software-in-the-loop and HIL tests (Wagner 2014)—"Model/Hardware/ Software-in-the-loop" block. Scrum CPS is also supported by sprints—design and hardware sprints—as well as by backlogs (Wagner 2014).

Agile product development is also proposed by Luedeke et al. to conduct CPS development, paired with Characteristics Properties Modelling/Property Driven Development (CPM/PDD) for product and process modelling (Baumeister et al., 2004), and with Design thinking for the creativity stage (Luedeke et al. 2018). CPM/PDD is based on Design thinking's output and is used in parallel with Agile product development. Design thinking is, from our understanding, related to human-centered design. This statement is supported by (Mueller and Thoring 2012). Given that Design thinking does not support product development but is instead a subset of it—the creativity and ideation stage—it is therefore classified as a method in this case. Moreover, the vocabulary employed by the authors for Agile product development—backlog and sprint—lead us to believe that Scrum is being used when talking about the Agile development process. Backlog and sprint are considered as tools to support the process.

Extreme Programming (XP), an Agile process, is also discussed by Escobar et al. (2017) for CPS development. XP is also supported by user stories, which are considered as a tool. Hence, an outline of an Agile *set* for CPS development is fragmented into different authors' works. Agile concepts and techniques are represented in Fig. 10.

3.3.5 Platform-, component- and contract-based design triptych

Zhu and Sangiovanni-Vincentelli introduce different frameworks—which are not related to one another—to support CPS development (Zhu and Sangiovanni-Vincentelli 2018). Beyond the frameworks, underlying concepts and techniques to support CPS development are mentioned, including platform-based design (PBD) (Davare et al. 2013; Sangiovanni-Vincentelli, Carloni, De Bernardinis & Sgroi, 2004), contract-based design theory, model-based and component-based design, as well as functional modelling.

From our understanding, PBD is a high-level method gathering engineering practices for system architecture and decomposition. PBD relies on the mapping between specifications and potential implementations through different abstraction layers. Each layer of abstraction is qualified as a design platform (Sangiovanni-Vincentelli et al. 2004; Nuzzo et al. 2015). PBD can be combined with contract-based design (Sangiovanni-Vincentelli et al. 2012; Seshia et al. 2017), which enables the reduction of design complexity. Contracts are defined as "mathematical models of the interface between components and levels of abstraction in a design" (Nuzzo et al., 2019). Thus, from our understanding, contract-based design is linked by Nuzzo et al. (2019) to component-based design and PBD via the levels of abstraction. In addition, contracts are considered as "an essential aspect of component-based design and interface theories" (Derler et al. 2013). Similarly in Sangiovanni-Vincentelli et al., although components are also discussed in relation to contract-based design, the link between component-based design and contract-based design is not formalized (Sangiovanni-Vincentelli et al. 2012). For complementary readings about contract theory and how it complements component-based design see Benveniste et al. (2015). Contracts can also be distinguished into vertical and horizontal contracts (Sangiovanni-Vincentelli et al. 2012; Nuzzo et al. 2015). Horizontal contracts—also known as traditional contracts—specify components' properties at one abstraction level and ensure their integrations' correctness, whereas vertical contracts are related to the refinement between two different abstraction levels in PBD (Sangiovanni-Vincentelli et al. 2012; Nuzzo et al. 2015). Hence, horizontal contracts are linked to components—interpreted as component-based design—and vertical contracts to PBD (Nuzzo et al. 2015). Nuzzo et al. also mention that PBD is based on a composition of components (Nuzzo et al. 2015). Thus, we map PBD with component-based design, and accordingly draft a triangle between PBD, component- and contract-based design.

Similarly, according to Seshia et al., PBD and contract-based design provide a framework for component-based design (Seshia et al. 2017). Although the authors do not expressly link component-based design with PBD and contract-based design, component-based design is seemingly mapped with both. In addition, Seshia et al. underscore the challenge of finding the right components, composition and contracts, interpreted as forming a trade-off triangle in between platform-, contract- and component-based design. Accordingly, beyond the formal links, the layout of platform-, contract- and component-based design can also be read as two different perspectives, resulting in a triangle. These two perspectives stem from (Sangiovanni-Vincentelli et al. 2012; Nuzzo et al. 2015; Seshia et al. 2017). First, a platform is based on a composition of components which are validated by contracts regarding desired properties; thereby supporting the idea that PBD is linked to component-based design, which in turn is mapped to contract-based design. Second, a platform through the contracts composes with verified components, accordingly linking PBD to contract-based design and subsequently to component-based design. The combination of both perspectives forms a triptych. Both perspectives are represented within the cartography.

Sangiovanni-Vincentelli et al. (2012) discuss multiple concepts and techniques, namely layered design to decompose systems' complexity "vertically", component-based design to reduce complexity "horizontally", the V-model used as a process, and model-based design that supports virtual integration. According to the authors, PBD encompasses horizontal (component-based design, virtual integration) and vertical (layered and model-based design) decompositions (Sangiovanni-Vincentelli et al. 2012). Component-based design also contributes to the reuse strategies represented as reuse engineering (Sangiovanni-Vincentelli et al. 2012; Seshia et al. 2017). Additionally, the V-model can also be supported by component-based design—as reuse can reduce the effort in both design and integration steps—and virtual integration (Sangiovanni-Vincentelli et al. 2012). The authors also enrich virtual integration with contract-based design—in this case, through the horizontal contract.

Complementary to the previous literature, component-based principles and practices are also addressed by Crnkovic, Malavolta, Muccini, and Sharaf (2016). Apparently linked with component-based practices, component-based modelling is proposed to "model and verify complex digital logic components in CPS" (Chen et al. 2015). It can be observed that component-based modelling differs from the other component-based techniques and is hence represented separately in the cartography.

In an automotive context, Wan et al. propose to transform functional model into architecture models that can be simulated and validated, thereby allowing the exploration of different architectures and design spaces. Functional modelling relies on the Functional Basis language and is organized according to a functional decomposition tree. Wan et al. consider functional modelling as part of an SE activity, mapping functional modelling to SE. Based on the functional model, the architecture is realized through architecture-based design—also referred to as PBD. The architecture is built using component libraries, which suggests the use of component-based design. In addition, the new components can be related to existing components and levels of abstraction through design contracts, which presumes the use of contract-based design, itself possibly linked to functional modelling (Wan et al. 2017).

These concepts and techniques are gathered on Fig. 11.

3.3.6 Other concepts and techniques

Apart from the other concepts and techniques, design for security and privacy was mentioned by Seshia et al. (2017), who underscore the necessity to integrate these aspects early in the product development process. Balasubramaniyan et al. propose a three-step "methodology": design, validation/simulation and verification. From our understanding, this methodology is closer to a method that conducts a technical procedure to achieve a compromise between stability and performance. This method includes the modelling of "timing imperfections" (Balasubramaniyan et al. 2016). Also connected to the inclusion of temporal constraints, Song et al. propose a three-step method composed of "functional modularization, networking for information-sharing, and coordination of system elements" (Song et al. 2019). This appears to be more related to a method, as it does not encompass the whole product development but only a subset. The first step focuses on functional decomposition into modules, the second identifies the delays that can occur and must be considered when designing the network, and the third step describes the information sharing and interactions between modules through unified interfaces. None of these three-step methods are named. They are represented on Fig. 11.

This section has presented an overview of the variety of concepts and techniques studied across the scientific literature to support CPS development. Based on the work presented in this section, a cartography has been established and is presented in the next section.

3.4 Cartography of concepts and techniques for cyber-physical systems development

The previous sub-sections listed the concepts and techniques recommended for CPS development. This section represents them and their links within a cartography based on the four-level model. Some concepts and techniques, as well as references, are added here to enhance the overall comprehension and structuration. As with mechatronics, CPS development gathers a large number of concepts and techniques that are then represented in two figures to improve clarity. Accordingly, Figs. 10 and 11 represent the cartography of concepts and techniques for CPS development. Figure 10 focuses on the core of SE and Agile practices, and Fig. 11 represents the platform-, contract- and component-based design triptych as well as miscellaneous concepts and techniques.

Compared to mechatronic product development, CPS development is a recent research area. The oldest reference discussing CPS development within the research results is from 2009. From an overall perspective, looking at.

Figure 10, the presence of an SE *set* can be observed, whose *backbone* is similar to that of mechatronic product development through the V-model, supported by model-based and model-driven practices, which are in turn supported by system modelling techniques. In the CPS development cartography, a smaller variety of approaches is also encountered, while only SE and Agile are discussed. Although the Agile approach is not explicitly mentioned, it can be implemented through the listed processes, methods, and tools, resulting in an incomplete Agile *set*. The process level also relies on only a few references, but the mentioned processes belong to the above-mentioned approaches. In this respect, the lack of references on the SE approach and on the V-model process limits a possible consensus on the SE *set* considered as a whole. However, it appears that the method level concentrates most of the studies, several of which integrate the use of models and are discussed next.

Similar to mechatronics, the method and tool levels integrate model-based and model-driven practices (such as model-based design, model-based development, MBSE, and model-driven architecture), as well as system modelling techniques. Model-based and model-driven practices, and system modelling techniques gather an important part of the research efforts, as represented in Fig. 10. This suggests a potential consensus on their use for CPS development. In addition, some authors explore other types of modelling. Among them, agent-based modelling (in Fig. 10) appears to be an ongoing subject. However, agent-based modelling remains a stand-alone block and needs to be further investigated to be integrated in a *set*, or at least mapped to other existing concepts and techniques.

Another group of methods is also gaining attention in the literature. As shown in Fig. 11, the platform-, component- and contract-based design triptych is supported by different references. However, this mapping is mainly developed at the method level and would need to be mapped to lower and upper levels. The link between component-based and contract-based design can be considered as an additional consensus. However, regarding the rest of the triptych, it should be noted that some references emerge from the same authors. Thus, the number of publications metric should be considered with some reserve. Still referring to methods, the user- and human-centered design method (mentioned in Fig. 10) also gathers multiple references and suggests that the human interaction within CPS needs to be considered. Indeed, some of these references refer to publications in medical CPS where the system is built around the users and interacts directly with them.

CPS development is strongly focused on architecture and modelling techniques rather than process and approach levels. Moreover, whereas mechatronics gathered some methods and tools from mechanical, control and software engineering, CPS concepts and techniques are apparently predominantly software-oriented. This statement supports the vision of Bricogne et al. (2016), who envisioned CPS as software-focused.

To synthesize, modelling is an acknowledged practice for managing the increasing complexity of CPS, as well as a foundation for CPS development. This modelling is mainly envisioned through model-based and model-driven practices using standardized languages, which are mostly depicted in Fig. 10. However, alternatives such as PBD with component- and contract-based design (Fig. 11), or agent-based modelling (Fig. 10), are also gaining research interest. Aside from a few works, the focus in the cartography seems to be on the methods. In addition, in Figs. 10 and 11, three structures of concepts and techniques can be observed: the SE *set*, the Agile incomplete *set* (Fig. 10), and the platform-, component-, and contract-based design triptych (Fig. 11). Currently, companies willing to develop CPS can rely on the proposed methods and tools and can possibly direct their product development practices towards modelling, and model-based and model-driven practices, which benefit from a possible consensus due to the sheer number of associated references.

In parallel to CPS, smart products are also the subject of a growing interest. These multidisciplinary products also need to be designed, and the next section explores the concepts and techniques for their development.

3.5 Concepts and techniques for smart product development

Smart products reflect the recent trend of product evolution, but with the integration of connectivity and the enhancement of IT. Smart products are surrounding us with various consumer products labelled as “smart”. This “smart” designation encompasses a wide variety of products, which make this term strongly marketing-connoted (Cronin 2010). However, the scientific literature has proposed different definitions. Among them, Porter and Heppelmann propose a general definition of “smart, connected products” composed of three core elements, which are physical, smart and connectivity components, justifying that smart products are multidisciplinary products (Porter and Heppelmann 2014). Tomiyama et al. (2019) also defend smart products as multidisciplinary products. For these researchers, smart products are "software intensive and the degree of multi-disciplinarity is much higher than, e.g., traditional mechatronics products, because the role of information is more than [that of] control" (Tomiyama et al., 2019). Rijsdijk and Hultink define smart products as "products that contain information technology (IT) in the form of, for example, microchips, software, and sensors and that are therefore able to collect, process, and produce information" (Rijsdijk and Hultink 2009). The scientific literature has also tried to position smart products relative to mechatronics and CPS (Abramovici, 2015; Anderl et al. 2013; Roblek et al. 2016; Tomiyama et al. 2019). Beyond the insoluble debate of the hierarchical layout between these concepts, our research interest focuses on the concepts and techniques used for the development of smart products and those that are common to both CPS and mechatronic product development.

First, according to Herzog and Bender, "the shift towards the paradigm of smart products also suggests that the current way of product development needs to be adapted fundamentally" (Herzog and Bender 2017). This recommendation is shared by Porter and Heppelmann, who underscore the necessity to implement new design principles, including software-based customization, personalization, and the integration of product upgrades and services (Porter and Heppelmann 2014). In addition, for the development of these smart products, the authors also recommend the development of an SE and agile software development expertise to integrate the different disciplines and their respective components (Porter and Heppelmann 2014). Ahram et al. also propose the use of SE, but from a process perspective (Ahram et al. 2011). Their process is based on a series of activities, including both project management and product development activities adapted from the Defense Acquisition University Guidebook. Some of these activities are supported by system modelling techniques through the use of SysML language.

Tomiyama et al. (2019) explore the development of smart products, and mainly defend the deployment of MBSE, which still may not be completely suitable for the reasons they present. System modelling techniques are also discussed, through the use of IDEF0, UML and SysML, as well as dynamic modelling. Hence, system modelling techniques support MBSE, which in turn supports the V-model which operationalizes an SE, building a *set*. However, Tomiyama et al. tend to qualify the use of the V-model as inappropriate because it is a prescriptive development model. They also list numerous concepts and techniques that can be used for smart product development, but which may be unsuitable in some cases. Among these techniques are Agile methods, modular design, Lean product development, Design for X, functional modelling, behaviour modelling, product architecture design, FMEA, QFD, and Kano. The authors also defend the need for a design for resilience (Tomiyama et al. 2019). Requirement engineering is cited as well, but is, from our understanding, considered as an engineering phase of product development rather than a concept or a technique.

Still on the approach level, Anderl et al. propose to make use of "Smart Engineering", which relies on a systems engineering extension, as connectivity and communication are enlarging the system borders (Anderl et al. 2013). The authors also emphasize that the V-model is not suitable for smart product development, since "approaches like the V-model do not provide systematic approaches to develop smart products' communication". Their proposed framework relies on their earlier work and an evolution of a V-model, the W-model (Nattermann and Anderl 2010), introduced for adaptronic systems. Accordingly, their work seems to propose a combination of an expanded approach and process to match smart product development features.

Rauch et al. propose to move towards a Lean product development (LPD) process supported by Industrie 4.0 technologies, concepts and solutions. Relying on the application of Axiomatic design (Suh 1998), the authors determine the Lean design parameters that are mapped with Industrie 4.0 technologies and concepts. The outcome is guidelines to provide a "Lean and Smart Product Development" for the development of smart products (Rauch et al. 2016). Nunes et al. also explore smart product development and the relation with Industrie 4.0 (Nunes et al. 2017). From their perspective, Industrie 4.0 and the associated technologies will foster the implementation of Lean principles and LPD; however, the authors do not explicitly suggest relying on LPD.

Miranda et al. (2017) underscore the importance of integrating environmental impact and sustainability. In that sense, the researchers propose to implement design for environment (DfE), Eco-design, sustainable design and lifecycle assessment (LCA). In addition, Miranda et al. introduce the "S^3 products", shorthand for the "sensing, smart and sustainable products" supported by a product model (Miranda, Pérez-Rodríguez, Borja, Wright & Molina, 2017). Based on the new product development (NPD) literature, their prior experience and the "integrated product, process and manufacturing system development reference model (IPPMD)", the authors propose a reference framework, "S^3 product development". Within the cartography, "S^3 product development" is represented at the conjunction of the IPPMD and NPD processes. In addition, an NPD process can be related to the stage-gate process (Cooper, 2004).

To support "S^3 product development", Miranda et al. (2017) propose a "toolbox" that gathers methods and tools to support each of the process steps. However, most of the concepts and techniques mentioned in their article are not supported by any references, which complicated their classification. For the ideation step, the researchers mention megatrends analysis, “A day in the life”, an empathy map, a job-to-be-done framework, outcome expectations, a matrix of needs and satisfiers, Kano, the analytic hierarchy process (AHP), a Pugh chart, value proposition, and storyboard. To support the “concept design and target specifications” step, Miranda et al. (2017) list several techniques: physical decomposition, functional decomposition, sustainable indicator repository, QFD, morphological matrix and structure, TRIZ, LCA, and FMEA. The AHP and Pugh charts are also mentioned for this step. For the detailed design step, DfX practices are proposed, including DfM, DfA and DfE, complemented by FMEA and LCA. Finally, to support the prototyping step, rapid prototyping, virtual prototypes and functional prototypes are listed. These prototyping techniques are represented within the “Virtual prototyping—3D modelling techniques” and “Physical and 3D Prototyping” blocks. FMEA is also mentioned for this step (Miranda, Pérez-Rodríguez, Borja, Wright & Molina, 2017).

To better exploit the digital era, Kim et al. propose the use of WebData as input for their process as a way to determine user requirements for smart products. The process is iterative and relies on QFD, modelling and simulation supported by finite element analysis, and finally on a physical prototype using rapid prototyping (Kim et al. 2018). The authors also integrate a user-centered mindset through the use of storytelling and user experience development. Their proposed process, called the “Smart product design-finite element analysis process (SPD-FEAP)”, has three stages, including feedback loops to support the development of user customized products. Smart products offer new possibilities through their data acquisition and connectivity capabilities, as the collected usage data can be reused for product development (Tomiyama et al. 2019). This data can be stored and managed thanks to the use of a digital twin (Tomiyama et al. 2019). The data collected are both “user-generated data” and “data serving as part of the product value” (Zheng et al. 2018). In that sense, Zheng et al. propose to rely on data-driven design for the co-development of smart products. Co-development is a collaborative development that combines the implication of stakeholders, such as manufacturers, partners and customers to generate new ideas and adapt smart products to customers’ needs (Zheng et al. 2018). Their proposed work seeks to implement the concept of customization and personalization of smart products, and to do so on both hardware and software aspects throughout the product lifecycle. Hence, the design process itself relies on a modular design stage—based on functional modelling—for the macro-level, and a scalable design stage for the micro-level. From our understanding, modular design supports co-development. However, this co-development lacks a formal description, and from our understanding is closer to an engineering practice than a complete process model to manage the complete product development. Hence, co-development is positioned here as a method. The modularity of smart products is also studied by Li, Roy & Saltz, (2017), who combine module-based (modular) and scale-based (scalable) design while also considering hardware and software modularity. Their proposition coordinates well with the evolving nature of smart products (Li, Roy & Saltz, 2017).

As with CPS, the digital era also brings out some functional safety and cybersecurity concerns for smart products. Axiomatic design and signal flow analysis (SFA) can be deployed to address these concerns (Riel et al. 2018). Within the automotive example, Riel et al. (2018) also discuss using the V-model process, but its relation with axiomatic design and SFA is not explicit (Riel et al. 2018). From our understanding, they use a subset of axiomatic design, positioning axiomatic design in this case, at the method level. SFA is related to signal flow graphs, grounded in electrical engineering (Abrahams and Coverley 1965), and is considered as a tool.

From an electronics engineering perspective, Crepaldi et al. (2014) make use of a top-down “constraint-driven methodology”. This method is based on the concept of constraint that is further explained in (Jerke, Lienig & Freuer, 2011). This constraint-driven methodology entails different tasks that are “constraint management and propagation, derivation, transformation and verification” (Crepaldi et al. 2014). From our perspective, the constraint-driven methodology is considered a method, as it translates high-level requirements into constraints, and it can be part of an overall development process.

Similar to mechatronics products, for design space exploration purposes, bond graphs can be used to model systems. Dagli et al. (2010) proposed the use of bond graphs, which can be transformed into SysML. As an object-oriented modelling, SysML is utilized to represent architectures that integrate the definition of interfaces. These architectures can then be optimized. Additionally, the same authors also propose an interface-based architecture development process that refines the requirements into functional, system and physical architectures. From our understanding, the interface-based architecture development process does not cover the whole development process, as it stops after the physical architecture. Hence, according to the decision tree, it is closer to the definition of a method.

Finally, in addition to “generic” smart products taken in their broad meaning, the development of specific smart products is also addressed in the literature. For example, regarding smart products dedicated to health, Rahimi and Ibarra insist on customer involvement, using their involvement to justify the exploration and use of user-centered design to develop these specific products (Rahimi

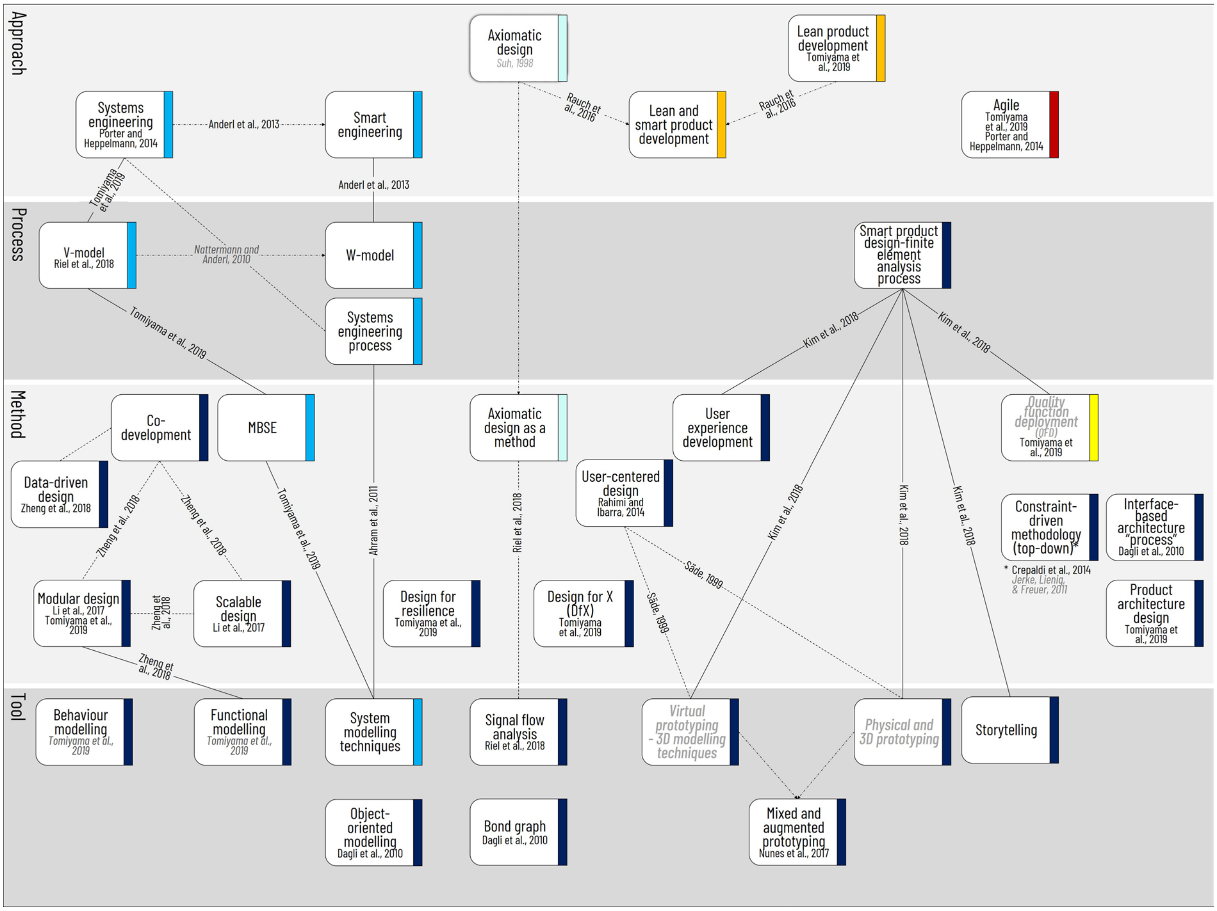

Fig. 12 Cartography of approaches, processes, methods and tools available for smart product development (1 of 2)

and Ibarra 2014). Linked with the user interaction and user-centered design techniques, Säde (1999) aims at supporting the development of usable smart products. Säde relies on modelling and prototyping through seven possible representations to assess both the hardware and software aspects of a product's usability. These seven tools are graphically represented through the blocks "Virtual prototyping—3D modelling techniques" and "Physical and 3D Prototyping". The representations describe "emerging products, [the] user interface and [the] interaction between user and product".

This section presented the variety of concepts and techniques proposed in the scientific literature to support smart product development. Based on the concepts and techniques presented here, a graphical representation has been established and is presented in the next section.

3.6 Cartography of concepts and techniques for smart product development

As with the previous cartographies, this section graphically represents the concepts and techniques recommended for smart product development, as well as their links within a cartography based on the four-level model. Some concepts and techniques are added to enhance the overall comprehension and structuration—such as the Stage-gate process, the Plan-driven—Systematic design approach, the NPD process, IPPMD and the Axiomatic design approach (Suh 1998). The cartography of concepts and techniques for smart product development is represented in Figs. 12 and 13. Figure 12 focuses on SE and miscellaneous concepts and techniques, and Fig. 13 focuses on Eco-design and "S^3 product development".

The graphical representation reveals that, contrary to CPS or mechatronics product development, smart product development is supported by scattered blocks at the different levels. Indeed, most of the cited concepts and techniques are only supported by single references. Hence, whereas CPS and mechatronic product development have a potential consensus emerging from reference aggregations, there is apparently little consensus on how to develop smart products. However, despite the absence of a consensus, smart product development can rely on the SE *set* mainly proposed by Tomiyama et al. (2019). That SE *set* encompasses the

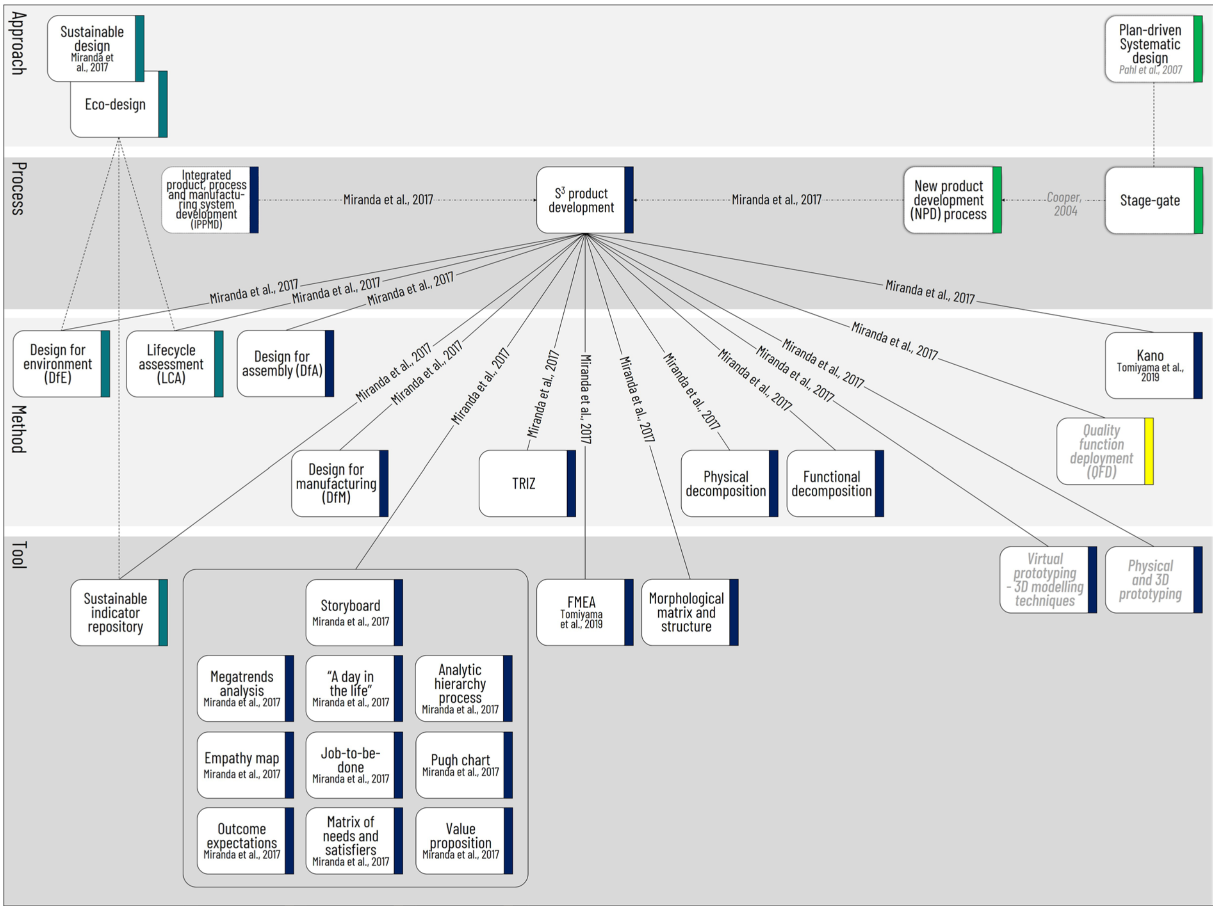
Approach
Sustainable design Miranda et al., 2017
Eco-design
Plan-driven Systematic design Pahl et al., 2007
Process
Integrated product, process and manufacturing system development (IPPMD)
Miranda et al., 2017
S^3 product development
Miranda et al., 2017
New product development (NPD) process
Cooper, 2004
Stage-gate
Method
Design for environment (DfE)
Lifecycle assessment (LCA)
Design for assembly (DfA)
Design for manufacturing (DfM)
TRIZ
Physical decomposition
Functional decomposition
Kano Tomiyama et al., 2019
Quality function deployment (QFD)
Miranda et al., 2017
Tool
Sustainable indicator repository
Storyboard Miranda et al., 2017
Megatrends analysis Miranda et al., 2017
"A day in the life" Miranda et al., 2017
Analytic hierarchy process Miranda et al., 2017
Empathy map Miranda et al., 2017
Job-to-be-done Miranda et al., 2017
Pugh chart Miranda et al., 2017
Outcome expectations Miranda et al., 2017
Matrix of needs and satisfiers Miranda et al., 2017
Value proposition Miranda et al., 2017
FMEA Tomiyama et al., 2019
Morphological matrix and structure
Virtual prototyping - 3D modelling techniques
Physical and 3D prototyping

Fig. 13 Cartography of approaches, processes, methods and tools available for smart product development (2 of 2)

V-model, as well as MBSE supported by system modelling techniques. Note that the presence of a single *set* does not indicate that it is the only way to develop a product; it instead suggests a consistent way to support product development across the different levels with compatible concepts and techniques.

Regarding the other approaches, Agile (Fig. 12), Eco-design and sustainable design (Fig. 13) are also addressed, as well as Lean product development and an adaptation called Lean smart product development (Fig. 12). With the exception of Eco-design, these approaches are not decomposed in the lower levels.

This overall picture of smart product development tends to confirm Anderl et al.'s statement regarding the fact that "although design methodology for mechatronic products is far advanced, a framework for developing Smart Products does not exist" (Anderl et al. 2013). In addition, Tomiyama et al. (2019) discuss some of the limitations of smart product development and their characteristics. These characteristics can make some of the methods and tools unsuitable for smart product development (Tomiyama et al., 2019). Accordingly, the literature introduces a variety of new frameworks, concepts and techniques, some relying, adapting and extending SE, LPD, IPPMD, or even QFD, with similar underlying concepts and principles. However, these derived—or labeled as "new"– concepts and techniques are generally standalone blocks and are rarely mapped to other existing concepts and techniques, potentially reducing their usability and their adoption by companies.

In addition, three observations must be mentioned. First, modular design is attracting some research interest, but it needs to be mapped within an overall process and approach. Modular design could help to both customize a product to the users' expectations and enable the products' reconfigurability. The second point is related to users and their integration throughout the product's development. From our understanding of the smart product development literature, user interaction is an important aspect of smart products (Kim et al. 2018; Rahimi and Ibarra 2014; Säde 1999; Tomiyama et al. 2019; Zheng et al. 2018). The third observation concerns Fig. 13, which depicts the toolbox proposed by Miranda et al. (2017) consisting of 24 methods and tools. This work stands out by offering more structuration than the rest of the smart product development landscape. However, the methods and tools mentioned all refer to steps of the "S^3 product development" process, but they are not structured together. In other words, the tools are directly attached to the process and do not support the methods.

In sum, the development of smart products seems less organized and less mature than that of mechatronics and CPS despite the similarities in the disciplines and expertise involved. Compared to CPS and mechatronics, there is no common agreement or orientation on how to develop smart products. Smart products are not a new type of product, but their definition has evolved over time, as have their features. Smart product development is expected to change accordingly. Indeed, smart products are now anticipated to be autonomous, resilient, and able to cope with changes in their environment. Hence, according to Tomiyama et al., "product design for smart products is not only about the product itself but also about [the] design of the mechanism for the product to change or grow". In addition, the collection of product usage and users' profile data will also influence product development, especially in the requirement definition phase (Tomiyama et al. 2019). The smart product development landscape is therefore expected to change with the introduction of new concepts and techniques.

3.7 Synthesis: multidisciplinary product development

To substantiate the conclusions drawn across the previous sections, two metrics are introduced to compare the cartographies of the three types of multidisciplinary products considered. The first metric (1) is linked to the density *d* of the mapping and is the number of solid lines divided by the number of concepts and techniques. Hence, as the density *d* tends to zero, its value can highlight the presence of isolated blocks. In contrast, a dense mapping—when density *d* tends to 1—indicates that a variety of concepts and techniques are more likely to be integrated and used together. The second metric (2) is related to the notion of consensus, noted as *c*. Consensus can be considered as the number of references divided by the number of concepts and techniques.

The metrics rely on data determined from the cartographies. Accordingly, *Nb* is the number of concepts and techniques—i.e., the number of blocks; *Nl* is the number of solid line links; and *Nr* is the number of references. Note that for the reference count, a reference on a solid line counts for each block and thus is counted twice; a reference on a dashed line is counted only once; and a reference on a hybridization link or an extended concept or technique is counted once as well (see Fig. 4). In addition, when several dashed lines converge to one block with a similar reference, the reference is only counted once, which lends the solid lines and their associated references a greater weight.

$$d = \frac{\mathrm{Nl}}{\mathrm{Nb}} \tag{1}$$

$$c = \frac{\mathrm{Nr}}{\mathrm{Nb}} \tag{2}$$

The mechatronics cartography includes 8 approaches, 19 processes, 46 methods and 52 tools, for a total of 125 concepts and techniques. These concepts and techniques are mapped by 106 formal links—i.e., solid lines. According to formula (1), the density *d* for mechatronics' mapping is 0.85. The CPS development is supported by 55 concepts and techniques distributed into 1 approach, 5 processes, 25 methods and 24 tools, all mapped by 41 formal links. The density *d* of CPS mapping is thus 0.75. Smart product development literature groups 56 concepts and techniques distributed into 6 approaches, 5 processes, 22 methods and 23 tools, mapped by 35 links. The resulting density *d* for smart product mapping is 0.63.

The mechatronic product development cartography contains 403 references distributed as 37 for the approach level, 75 for the process level, 153 for the method level, and 138 for the tool level. Hence, the overall consensus *c* for mechatronic product development is 3.2. The CPS cartography gathers a total of 184 references distributed as 4 for the approach level, 19 for the process level, 103 for the method level and 58 for the tool level, resulting in an overall consensus value *c* for CPS development of 3.3. The smart product development cartography relies on 104 references, distributed as 8 for the approach level, 36 for the process level, 31 for the method level, and 29 for the tool level. Hence, the overall consensus *c* for smart product development is 1.9.

The consensus *c* value is also calculated according to formula (2) for each level. The results are represented in Table 1. The consensus value per level indicates which level attracts the most research interest and thus, which level is more likely to present a consensus suggested by multiple references.

Considering these results, it can be observed that mechatronic product development benefits from the overall

Table 1 Calculation of overall density, overall consensus, and consensus per level for mechatronic, CPS and smart product development

Metrics	Mechatronic product dev	CPS product dev	Smart product dev
$d_{overall}$	0.85	0.75	0.63
$c_{overall}$	3.2	3.3	1.9
$c_{approach}$	4.6	4	1.3
$c_{process}$	3.9	3.8	7.2
c_{method}	3.3	4.1	1.4
c_{tool}	2.7	2.4	1.3

highest density value, followed by CPS and smart product development. In terms of consensus, CPS development benefits from the overall highest value, followed by mechatronics and smart product development.

Looking at mechatronic product development, we can observe that the consensus per level is higher for the approach and process levels. In contrast, the tool level value is lower and tends to lower the overall consensus value. These numbers tend to substantiate the observations made in Section 3.2. The cartography reveals a significant number of references gathered around the SE *set's backbone*—encompassing SE, V-model, model-based and model-driven practices, and system modelling techniques (see Fig. 6). At the tool level, the different modelling techniques gather a large number of references, but these numbers are countered by numerous single-reference blocks, explaining the lower consensus for the tool level. Moreover, some of these blocks are isolated, which impacts the density as well.

Regarding CPS development, the consensus per level analysis tends to confirm the fact that the literature focuses on the method level. This can be explained by the many references linked to model-based and model-driven practices, as well as the triptych of PBD, component- and contract-based design. Hence, the method level of CPS offers a higher consensus value than the mechatronics and smart products method levels. However, as for mechatronics, the tool level tends to lower the overall consensus value for similar reasons—the presence of isolated and single-reference blocks. Finally, the consensus value for the approach and process levels are relatively high in the context of Table 1. This reflects the small number of approaches and processes, combined with the weights accorded to references on a solid line—for example, see "Scrum CPS" in Fig. 10. However, on the cartography, these approach and process levels do not generate a consensus.

For smart product development, the consensus values at the approach, method, and tool levels are, in the context of Table 1, relatively low, which, combined with the lowest density, substantiates our observations of a greater number of isolated and single-reference blocks. Meanwhile, the consensus value at the process level contrasts sharply with the other values in Table 1. This value is explained by the fact that the "S^3 product development" process proposed by Miranda et al. (2017) is linked by solid lines to the different techniques of the proposed toolbox. Therefore, the block is endorsed by a reference that is counted several times, thereby increasing the value of the metric. Actually, the process level of smart product development does not have a consensus on the use of "S^3 product development", because only Miranda et al. (2017) mention it.

Finally, the metrics corroborate the previous cartographies' synthesis. The overall consensus and density metrics tend to moderate the variations between the levels, while highlighting the gap between the development of mechatronics and CPS on the one hand, and the development of smart products on the other. Mechatronic product development and CPS development combine a comparatively high consensus with a dense mapping, enabling them to support companies in their adoption of new practices. Companies can effectively rely on the scientific literature and deploy SE or Agile concepts and techniques, especially model-based, model-driven and modelling practices. The case of smart product development is quite different. The mapping density is lower and there is still no consensus that could guide companies in the adoption of new practices at any level.

4 The multidisciplinary product development landscape

This paper presents a survey encompassing the development of three types of multidisciplinary products, graphically represented through cartographies. The different cartographies rely on the four-level model to group and sort the variety of concepts and techniques available to support each MPD. The proposed cartographies and analysis can benefit both research and industry.

4.1 From a research perspective: an overall perspective and analysis of MPD

For the research domain, the work presented here provides a global outlook to other researchers who aspire to position their contribution with respect to the existing literature. It highlights possible gaps in product development literature and helps to direct further research. Moreover, the established cartographies could eventually be further analysed to determine the different research trends and directions regarding MPD over time. Indeed, this analysis could be done by examining the publication years, which could identify evolutions in the proposed concepts and techniques and thereby reveal some "trends". In addition, given the increasing movement towards sustainability within society, eco-design has been studied in more depth recently by a handful of authors (Favi et al. 2019; Merschak and Hehenberger 2019). Continuing research on sustainability could help to resolve the issue of moving forward with technological development within the fourth industrial revolution while supporting eco-responsible products.

4.2 From an industrial perspective: enabling the adoption of concepts and techniques

From a practical standpoint, combined with an in-depth analysis, the cartography sheds light on the maturity of mechatronics, CPS and smart product development. In turn, the cartography enables companies to identify new or untapped engineering practices and fosters the adoption of concepts and techniques for complex product development. Each level of the cartography can relate to organisational levels within companies. These organisational levels can be labelled as strategic, tactical and operational levels (Girard and Doumeingts 2004). The approaches can be envisioned as corresponding to the strategic level, the processes to the tactical level and methods and tools to the operational level.

Multidisciplinary products are acknowledged for their complexity and the organisational complexity they engender. This complexity can itself justify the deployment of a SE *set* of concepts and techniques with a focus on the usage of modelling practices. Various authors across the three types of products defend the use of this *set*. However, the success of these multidisciplinary products can also be dependent on the user experience. Users have evolving needs and expectations. This perspective on products is addressed by Agile and user-centered techniques, as well as some quality methods such as QFD. However, complexity and user experience seem to be two distinct focuses in engineering, which can be managed by SE and Agile practices, respectively. This could raise the issues of how to manage high complexity in Agile and how to make SE more "agile". In fact, within the cartographies, Agile and SE practices have only a few links. From an organisational perspective, Agile and SE are also opposed; with the structured and systematic aspect of SE on one hand, and the lightweight and attractive flexibility of Agile on the other. These different questions have been partially addressed in the literature. Hence, a worthwhile research perspective would be to further explore the hybridization of Agile and SE, as proposed by (Stelzmann 2012; Mabrouk et al. 2018; Plateaux et al. 2020) and represented by the purple rectangle in the cartographies.

Given that CPS and mechatronic product development share common concepts and techniques, this raises the question of the differences between CPS and mechatronics. Indeed, if both are developed according to a similar focus on modelling, SE, and Agile practices, the question of their intrinsic differences arises. Thus, the impact on design practitioners of this distinction between CPS and mechatronics remains to be delineated. From our understanding, within CPS there are major concerns about non-functional properties such as security, reliability, robustness and resilience linked with the introduction of connectivity and a changing environment. Some examples of these concerns are addressed in (Woo et al. 2008; Hughes and Cybenko 2014; Vachtsevanos et al. 2018; Yuan and Xia 2018; O'Halloran et al. 2018; Brooks and Roy 2021). However, these concerns are not explicitly reflected within the cartography, as few methods and tools are dedicated to support them, and some authors have merged these concerns within existing methods and tools.

There is a large range of possibilities regarding the concepts and techniques to be deployed—over 200. However, not all of them should be considered with the same importance, since not all of the proposed works have been studied equally. Concepts and techniques supported by multiple authors are more likely to achieve a consensus within the scientific community on how to develop a product. Moreover, the question of the validity of 25-year-old work in the current context could also be discussed. Hence, to better guide companies in choosing from the possible concepts and techniques to adopt, the question of the selection remains to be explored.

4.3 The limitations: suggested improvements

The cartographies aggregate a large variety of concepts and techniques, but they do not indicate how to select or adapt these new concepts and techniques. Some studies have already focused on the selection of development processes, methods and tools, such as the work proposed by (Buchert et al. 2017; Goevert and Lindemann 2018). The contextualisation of the project and the company is likely to be a first step towards envisioning a selection. Indeed, the selection can rely on intrinsic features of the products to develop, the project and company's environment as well as their designers' skills. Accordingly, Tomiyama et al. stated "while it is absolutely impossible to state which one is the best, the choice depends on the application and designer's skill and experiences" (Tomiyama et al. 2009). Hence, further investigations should be conducted on how to select the appropriate *set*, or complementary concepts and techniques based on the established cartographies. The cartography can be envisioned as a validated database of concepts and techniques endorsed by scientific literature which must be kept up to date and maintained to remain useful.

Furthermore, the established cartographies and their in-depth analysis are the result of a significant research effort to represent a comprehensive view of the MPD landscape based on the scientific literature. However, the presented work relies on static representations and can appear as a snapshot. To overcome this limitation, a possible improvement would be to transfer these cartographies into the form of a collaborative ontology to allow a dynamic and navigable representation. This ontology could enable researchers and companies to contribute to the evolution of MPD, realizing a community-based and maintained database. The ontology classes could be defined according to the four-level model, and the addition of concepts and techniques could be supervised to preserve the overall coherence. Moreover, the ontological representation would enable companies to request the ontology through defined filters and thereby support their selection of concepts and techniques.

Another potential limitation of the presented work is that consensus is drawn from the number of references. However, this metric does not reflect the variety of authors. For instance, a laboratory publishing intensively on a new concept or a technique can appear as a consensus, while consensus exists when both high number and variety are met. One such example within the cartographies is the work from Sangiovanni-Vincentelli on the PBD, contract- and component-based triptych.

Another point is related to the level of interpretation. In fact, within the cartographies, the mapping and the position of the blocks rely on the content of the articles and the authors' interpretations. As way to meaningfully limit the interpretations, the decision tree presented in Fig. 3 aims at ensuring a systematic classification of the different concepts and techniques. For the specific cases or when different interpretations are possible, the authors developed their own argumentation. Regarding the mapping and the links, this point has been addressed with the introduction of dashed and solid lines, making a distinction between a stated link and an interpreted link. In addition, in the different metrics, a higher importance was attributed to solid lines.

Finally, the database queries were limited to journals and article titles. A possible improvement would be to address the other meta-data such as the keywords and the abstracts to gather complementary studies. Finally, the different queries could be extended to conference papers, resulting in a deeper view of MPD and enriched cartographies.

5 Conclusion

Through the integration of new technologies within products, the fourth industrial revolution and its associated digital and connectivity era create new challenges for their development. Accordingly, companies are invited to reconsider and adopt new concepts and techniques to support MPD. MPD is documented within the scientific literature through the development of mechatronics, CPS and smart products, which are three types of multidisciplinary products. After defining each of these multidisciplinary products, the authors surveyed the available concepts and techniques to support their development.

A total of 236 concepts and techniques were identified from 167 scientific publications. Of these 236 concepts and techniques, 125 are proposed for mechatronic product development, 55 for CPS development, and 56 for smart product development, with many overlaps. These concepts and techniques were classified with the help of a proposed four-level model and the associated decision tree, either as an approach, a process, a method or a tool. These concepts and techniques were organized and presented within cartographies, one for each type of product.

The contribution of this paper is threefold. First, the presented work offers a comprehensive view of MPD, through the identification of the approaches, processes, methods and tools that can support mechatronics, CPS or smart product development. The four-level model organizes the 236 concepts and techniques, connected by 182 links. Second, to the authors' knowledge, this is the only work to concurrently address mechatronics, CPS and smart product development; thereby allowing their differences and commonalities to be distinguished based on their respective cartography and the calculated metrics. Third, the cartographies organize the fragmented landscape of multidisciplinary product development, and thus help to navigate the dense body of knowledge from the scientific literature. This work can also help researchers to position their future work in a broader perspective as well as support companies in the transformation of their product development practices in line with the fourth industrial revolution.

The analysis shows that mechatronic product development relies on a well-defined foundation with the SE *set*, linking the V-model, model-based and model-driven practices, and system modelling techniques. Agile also collects processes, methods and tools, but their discontinued links prevent the formation of a *set*. Similarly, axiomatic design and systematic design are discussed but do not aggregate a *set*. CPS development shares similarities with mechatronics, especially the consensus on model-based and model-driven practices, system modelling techniques, as well as the fact that SE and Agile are a focus in both. In addition, CPS development can rely on a triptych of methods involving platform-, component- and contract- based design. Finally, smart product development, along with the development of CPS and mechatronic products, can be conducted with the SE *set*. However, to date, few references support this *set*, and so we could not consider it as a consensus. In general, literature for smart product development is composed of different scattered blocks. Hence, smart product development appears to have no common agreement or common direction.

From all our results it is possible to discern a potential direction for companies willing to evolve their engineering practices for MPD. The SE *set* appears to be the most developed among the three types of product development, and it integrates model-based and model-driven practices and system modelling techniques. Those practices and techniques allow it to manage the increasing complexity, foster collaboration, and enable simulations. Accordingly, SE *set* appears as a preferred—but not unique—option to support MPD. These observations are substantiated by the density and consensus metrics.

Finally, the next step in our research work would be to assess how companies are currently preparing for the fourth industrial revolution from a product development perspective. An empirical study is envisioned to better understand how companies are developing multidisciplinary products and to compare these results to the cartographies based on the common use of the four-level model.

Acknowledgements This work has been funded by France's "Fonds Unique Interministériel—FUI19 programme" and assisted by System@TIC and i-Trans competitiveness clusters (the MIMe project). The authors are grateful to all the consortium partners for their valuable contributions during the development of ideas and concepts.

Funding This work has been funded by France's "Fonds Unique Interministériel—FUI19 programme" and assisted by System@TIC and i-Trans competitiveness clusters (the MIMe project).

Data availability A searchable version of the cartographies is available at (Guérineau et al. 2022).

Code availability Not applicable.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

Abrahams JR, Coverley GP (1965) Signal flow analysis. Pergamon Press

Abramovici M (2015) Smart products. CIRP encyclopedia of production engineering. Springer, Heidelberg, Berlin, pp 1–5

Abramovici M, Bellalouna F (2007) Integration and complexity management within the mechatronics product development. Advances in life cycle engineering for sustainable manufacturing businesses. Springer, London, pp 113–118

AFNOR (2008) NF E01–010 Mécatronique—Vocabulaire, AFNOR

Ahram T, Karwowski W, Amaba B (2011) Collaborative systems engineering and social-networking approach to design and modelling of smarter products. Behav Inf Technol 30:13–26. https://doi.org/10.1080/0144929X.2010.490957

Ahrens M, Richter C, Hehenberger P, Reinhart G (2018) Novel approach to establish model-based development and virtual commissioning in practice. Eng Comput 0:1–14. https://doi.org/10.1007/s00366-018-0622-6

Akkaya I, Derler P, Emoto S, Lee EA (2016) Systems engineering for industrial cyber-physical systems using aspects. Proc IEEE 104:997–1012. https://doi.org/10.1109/JPROC.2015.2512265

Albers A, Reiß N, Bursac N, et al (2014) Situation-appropriate method selection in product development process - empirical study of method application. In: Proc Nord 2014 Conf Nord 2014, Aalto Design Factory, Aalto, pp 550–559

Alur R (2015) Principles of Cyber-physical Systems. MIT Press

Alvarez Cabrera AA, Foeken MJ, Tekin OA et al (2010) Towards automation of control software: a review of challenges in mechatronic design. Mechatronics 20:876–886. https://doi.org/10.1016/j.mechatronics.2010.05.003

Alvarez Cabrera AA, Woestenenk K, Tomiyama T (2011) An architecture model to support cooperative design for mechatronic products: a control design case. Mechatronics 21:534–547. https://doi.org/10.1016/j.mechatronics.2011.01.009

Amuthakkannan R (2012) Parameters design and performance analysis of a software-based mechatronics system using Taguchi robust design—a case study. Int J Product Qual Manag 10:1–24. https://doi.org/10.1504/ijpqm.2012.047939

Anderl R (2014) Industrie 4.0 - Advanced Engineering of Smart Products and Smart Production. In: 19th Int Semin High Technol, pp 1–14. https://doi.org/10.13140/2.1.1039.4406

Anderl R, Picard A, Albrecht K (2013) Smart Engineering for Smart Products Reiner. In: Abramovici M, Stark R (eds) Smart Product Engineering. Lecture Notes in Production Engineering. Springer Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-30817-8_1

Araz M, Erden Z (2014) Behavioural representation and simulation of design concepts for systematic conceptual design of mechatronic systems using Petri Nets. Int J Prod Res 52:563–583. https://doi.org/10.1080/00207543.2013.838648

Attarzadeh-Niaki SH, Sander I (2016) An extensible modeling methodology for embedded and cyber-physical system design. SIMULATION 92:771–794. https://doi.org/10.1177/0037549716659753

Bachmann V, Messnarz R (2012) Improving the software development for multiple projects by applying a platform strategy for mechatronic systems. J Softw Evol Process 24:541–549. https://doi.org/10.1002/smr.495

Baheti R, Gill H (2011) Cyber-Physical Systems. Impact Control Technol 1:161–166. https://doi.org/10.1145/1795194.1795205

Balasubramaniyan S, Srinivasan S, Buonopane F et al (2016) Design and verification of cyber-physical systems using TrueTime, evolutionary optimization and UPPAAL. Microprocess Microsyst 42:37–48. https://doi.org/10.1016/j.micpro.2015.12.006

Barbieri G, Derler P, Auslander DM, et al (2016) Design of mechatronic systems through aspect and object-oriented modeling. Autom 64:244–252. https://doi.org/10.1515/auto-2015-0071

Barbieri G, Fantuzzi C, Borsari R (2014) A model-based design methodology for the development of mechatronic systems. Mechatronics 24:833–843. https://doi.org/10.1016/j.mechatronics.2013.12.004

Bauernhansl T (2014) Die Vierte Industrielle Revolution—Der Weg in ein wertschaffendes Produktionsparadigma. In: Bauernhansl T, ten Hompel M, Vogel-Heuser B (eds) Industrie 4.0 in Produktion, Automatisierung und Logistik. Springer Fachmedien, Wiesbaden, pp 5–35

Baumeister H, Knapp A, Wirsing M (2004) Property-driven development. Proc Second Int Conf Softw Eng Form Methods SEFM 2004:96–102. https://doi.org/10.1109/SEFM.2004.1347507

Beck K, Beedle M, Van Bennekum A, et al (2001) Agile Manifesto. http://agilemanifesto.org/iso/fr/principles.html. Accessed 28 Jun 2016

Behbahani S, de Silva CW (2007) Reliability tradeoffs of a complex mechatronic system in the early design stage. Int J Manuf Res 2:51–73. https://doi.org/10.1504/ijmr.2007.013426

Benveniste A, Caillaud B, Nickovic D, et al (2015) Contracts for Systems Design : Theory, Research report N°8759, Inria

Blanchard BS, Fabrycky W (1990) Systems Engineering and Analysis, 4th edn. Prentice Hall

Bloem J, van Doorn M, Duivestein S, et al (2014) The Fourth Industrial Revolution Things to Tighten the Link Between IT and OT, Sogeti VINT Report.

Böhmer AI (2018) When digital meets physical—Agile Innovation of Mechatronic Systems. Dissertation, Technische Universität München

Bricogne M (2015) Méthode Agile pour la conception collaborative multidisciplinaire de systèmes intégrés : application à la Mécatronique. Dissertation, Université de Technologie de Compiègne

Bricogne M, Le Duigou J, Eynard B (2016) Design Processes of Mechatronic Systems. In: Bradley D, Hehenberger P (eds) Mechatronic Futures. Springer International Publishing, pp 75–89. https://doi.org/10.1007/978-3-319-32156-1_6

Brooks S, Roy R (2021) An overview of self-engineering systems. J Eng Des 32:397–447. https://doi.org/10.1080/09544828.2021.1914323

Broy M, Schmidt A (2014) Challenges in engineering cyber-physical systems. Computer 47:70–72. https://doi.org/10.1109/MC.2014.30

Brunetti G, Golob B (2000) Feature-based approach towards an integrated product model including conceptual design information. CAD Comput Aided Des 32:877–887. https://doi.org/10.1016/S0010-4485(00)00076-2

Buchert T, Halstenberg FA, Bonvoisin J et al (2017) Target-driven selection and scheduling of methods for sustainable product development. J Clean Prod 161:403–421. https://doi.org/10.1016/j.jclepro.2017.05.067

Bures T, Matena V, Mirandola R, et al (2018) Performance Modelling of Smart Cyber-Physical Systems. In: Companion of the 2018 ACM/SPEC International Conference on Performance Engineering—ICPE '18. ACM Press, New York, USA, pp 37–40. https://doi.org/10.1145/3185768.3186306

Burmester S, Giese H, Tichy M (2005) Model-Driven Development of Reconfigurable Mechatronic Systems with Mechatronic UML. In: Aßmann U, Aksit M, Rensink A (eds) Model Driven Architecture, MDAFA 2004, MDAFA 2003, Lecture Notes in Computer Science. Springer Heidelberg, Berlin, pp 47–61. https://doi.org/10.1007/11538097_4

Buur J, Andreasen MM (1989) Design models in mechatronic product development. Des Stud 10:155–162. https://doi.org/10.1016/0142-694X(89)90033-1

Cagan J, Vogel MV (2002) Creating Breakthrough Products. Financial Times / Prentice Hall

Cao Y, Liu Y, Paredis CJJ (2011) System-level model integration of design and simulation for mechatronic systems based on SysML. Mechatronics 21:1063–1075. https://doi.org/10.1016/j.mechatronics.2011.05.003

Carni DL, Grimaldi D, Lamonaca F et al (2017) From distributed measurement systems to cyber-physical systems: a design approach. Int J Comput 16:66–73. https://doi.org/10.1109/I2MTC.2017.7969977

Chami M, Bruel JM (2015) Towards an integrated conceptual design evaluation of mechatronic systems: The SysDICE approach. Procedia Comput Sci 51:650–659. https://doi.org/10.1016/j.procs.2015.05.180

Chan LW, Leung TP (1996) Spiral design model for consumer mechatronic products. Mechatronics 6:35–51. https://doi.org/10.1016/0957-4158(95)00041-0

Chen CY, Chen IM, Cheng CC (2009) Integrated design of legged mechatronic system. Front Mech Eng China 4:264–275. https://doi.org/10.1007/s11465-009-0060-7

Chen F, Ye H, Yang J et al (2015) A standardized design methodology for complex digital logic components of cyber-physical systems. Microprocess Microsyst 39:1245–1254. https://doi.org/10.1016/j.micpro.2015.08.012

Chen R, Liu Y, Zhao J, Ye X (2019) Model verification for system design of complex mechatronic products. Syst Eng 22:156–171. https://doi.org/10.1002/sys.21470

Clarkson J, Eckert C (2005) Design process improvement. Springer, London

Cooper RG (2004) New Products—What Separates the Winners from the Losers and What Drives Success. In: Kenneth B, Kahn KB (eds) The PDMA Handbook of New Product Development. John Wiley & Sons, Inc., pp 3–28. https://doi.org/10.1002/9780470172483.ch1

Counsell J, Porter I, Dawson D, Duffy M (2002) Schemebuilder: computer aided knowledge based design of mechatronic systems. Assem Autom 19:129–138. https://doi.org/10.1108/01445159910265206

Couturier P, Lô M, Imoussaten A et al (2014) Tracking the consequences of design decisions in mechatronic Systems Engineering. Mechatronics 24:763–774. https://doi.org/10.1016/j.mechatronics.2014.03.004

Craig KC, Stolfi FR (1994) Introduction to Mechatronic System Design with Applications: Course Book. IEEE

Crepaldi M, Grosso M, Sassone A et al (2014) A top-down constraint-driven methodology for smart system design. IEEE Circuits Syst Mag 14:37–57. https://doi.org/10.1109/MCAS.2013.2296415

Crnkovic I, Asklund U, Dahlqvist AP (2003) Implementing and Integrating Product Data Management and Software Configuration Management. Artech House, Boston & London

Crnkovic I, Malavolta I, Muccini H, Sharaf M (2016) On the Use of Component-Based Principles and Practices for Architecting Cyber-Physical Systems. In: Proc - 2016 19th Int ACM SIGSOFT Symp Component-Based Softw Eng (CBSE), pp 23–32. https://doi.org/10.1109/CBSE.2016.9

Cronin MJ (2010) Smart Products. Cambridge University Press, Cambridge, Smarter Services

Dagli CH, Singh A, Dauby JP, Wang R (2010) Smart systems architecting: computational intelligence applied to trade space exploration and system design. Syst Res Forum 03:101–119. https://doi.org/10.1142/s1793966609000109

Danilovic M, Browning TR (2007) Managing complex product development projects with design structure matrices and domain mapping matrices. Int J Proj Manag 25:300–314. https://doi.org/10.1016/j.ijproman.2006.11.003

Danjou C, Rivest L, Pellerin R (2017) Industrie 4.0 : des pistes pour aborder l'ère du numérique et de la connectivité, CEFRIO

Darwish A, Hassanien AE (2018) Cyber physical systems design, methodology, and integration: the current status and future outlook. J Ambient Intell Humaniz Comput 9:1541–1556. https://doi.org/10.1007/s12652-017-0575-4

Dauphin-Tanguy G (2008) Integrated design of mechatronic systems. In: AIP Conference Proceedings. In: AIP Conference Proceedings, pp 574–574. https://doi.org/10.1063/1.2953049

Davare A, Densmore D, Guo L et al (2013) metro II: a design environment for cyber-physical systems. ACM Trans Embed Comput Syst 12:1–31. https://doi.org/10.1145/2435227.2435245

Delbecq S, Budinger M, Hazyuk I et al (2017) A framework for sizing embedded mechatronic systems during preliminary design. IFAC-PapersOnLine 50:4354–4359. https://doi.org/10.1016/j.ifacol.2017.08.875

Derelöv M, Detterfelt J, Björkman M, Mandenius CF (2008) Engineering design methodology for bio-mechatronic products. Biotechnol Prog 24:232–244. https://doi.org/10.1021/bp0701822

Derler P, Lee EA, Torngren M, Tripakis S (2013) Cyber-physical system design contracts. In: 2013 ACM/IEEE Int Conf Cyber-Physical Syst ICCPS 2013, pp 109–118. https://doi.org/10.1109/ICCPS.2013.6604005

Dieterle W (2005) Mechatronic systems: automotive applications and modern design methodologies. Annu Rev Control 29:273–277. https://doi.org/10.1016/j.arcontrol.2005.05.002

Dimitrov V, Jagtap V, Skorinko J et al (2015) Human-centered design of a cyber-physical system for advanced response to Ebola (CARE). Proc Annu Int Conf IEEE Eng Med Biol Soc EMBS 2015:6856–6859. https://doi.org/10.1109/EMBC.2015.7319968

Drath R, Horch A (2014) Industrie 4.0: Hit or hype? IEEE Ind Electron Mag 8:56–58. https://doi.org/10.1109/MIE.2014.2312079

Dubberly H (2004) How do you design? Duberly Design Office, San Francisco

Dupuis JF, Fan Z, Goodman E (2015) Evolutionary design of discrete controllers for hybrid mechatronic systems. Int J Syst Sci 46:303–316. https://doi.org/10.1080/00207721.2013.783643

Ebert C (2013) Improving engineering efficiency with PLM/ALM. Softw Syst Model 12:443–449. https://doi.org/10.1007/s10270-013-0347-3

Ebert C, Paasivaara M (2017) Scaling Agile. IEEE Softw 34:98–103. https://doi.org/10.1109/MS.2017.4121226

Erol S, Schumacher A, Sihn W (2016) Strategic guidance towards Industry 4.0 – a three-stage process model. In: Int Conf Compet Manuf, pp 495–501

Escobar L, Carvajal N, Naranjo J, et al (2017) Design and implementation of complex systems using Mechatronics and Cyber-Physical Systems approaches. In: 2017 IEEE Int Conf Mechatronics Autom ICMA 2017, pp 147–154. https://doi.org/10.1109/ICMA.2017.8015804

Fan H, Liu Y, Hu B, Ye X (2016) Multidomain model integration for online collaborative system design and detailed design of complex mechatronic systems. IEEE Trans Autom Sci Eng 13:709–728. https://doi.org/10.1109/TASE.2015.2390039

Fan Z, Wang J, Goodman E (2004) Exploring open-ended design space of mechatronic systems. Int J Adv Robot Syst 1:295–302. https://doi.org/10.5772/5636

Farid AM, Ribeiro L (2015) An axiomatic design of a multiagent reconfigurable mechatronic system architecture. IEEE Trans Ind Informatics 11:1142–1155. https://doi.org/10.1109/TII.2015.2470528

Favi C, Marconi M, Germani M, Mandolini M (2019) A design for disassembly tool oriented to mechatronic product de-manufacturing and recycling. Adv Eng Informatics 39:62–79. https://doi.org/10.1016/j.aei.2018.11.008

Feng Y, Hu B, Hao H et al (2018) Design of distributed cyber-physical systems for connected and automated vehicles with implementing methodologies. IEEE Trans Ind Informatics 14:4200–4211. https://doi.org/10.1109/TII.2018.2805910

Fitzgerald J, Gamble C, Larsen PG, et al (2015) Cyber-Physical Systems Design: Formal Foundations, Methods and Integrated Tool Chains. In: Proc—3rd FME Work Form Methods Softw Eng Formalise 2015, pp 40–46. https://doi.org/10.1109/FormaliSE.2015.14

Forsberg K, Mooz H (1991) The relationship of system engineering to the project cycle. INCOSE Int Symp 1:57–65. https://doi.org/10.1002/j.2334-5837.1991.tb01484.x

freddi D (2009) The integration of old and new technological paradigms in low- and medium-tech sectors: the case of mechatronics. Res Policy 38:548–558. https://doi.org/10.1016/j.respol.2008.10.017

French MJ (1985) Conceptual design for engineers. Springer, Heidelberg, Berlin. https://doi.org/10.1007/978-3-662-11364-6

Fricker SA (2012) Software Product Management. In: Maedche A, Botzenhardt A, Neer L (eds) Software for People, Management for Professionals. Springer Heidelberg, Berlin, pp 53–81. https://doi.org/10.1007/978-3-642-31371-4_4

Fruchter R, Reiner KA, Toye G, Leifer LJ (1996) Collaborative Mechatronic System. Design Concurr Eng 4:401–412. https://doi.org/10.1177/1063293X9600400408

Fu Y, Choosilp W, Dong Z (2018) Model-based test-driven cyber-physical system design. In: SoutheastCon 2018. IEEE, pp 1–6. https://doi.org/10.1109/SECON.2018.8479080

Gamage LB, De Silva CW, Campos R (2011) Design evolution of mechatronic systems through modeling, on-line monitoring, and evolutionary optimization. Mechatronics 22:83–94. https://doi.org/10.1016/j.mechatronics.2011.11.012

Gao Z, Xia H, Dai G (2011) A model-based software development method for automotive cyber-physical systems. Comput Sci Inf Syst 8:1277–1301. https://doi.org/10.2298/csis110303059g

García-Valls M, Perez-Palacin D, Mirandola R (2018) Pragmatic cyber physical systems design based on parametric models. J Syst Softw 144:559–572. https://doi.org/10.1016/j.jss.2018.06.044

Gausemeier J, Dumitrescu R, Kahl S, Nordsiek D (2011) Integrative development of product and production system for mechatronic products. Robot Comput Integr Manuf 27:772–778. https://doi.org/10.1016/j.rcim.2011.02.005

Gendron S, Brousseau J, Elouafi A, Urli B (2011) Development of a Tool for Assessing the Design Process: A step toward the creation of a Design Observatory. In: Proceedings of the Canadian Engineering Education Association. https://doi.org/10.24908/pceea.v0i0.3626

Gericke K, Blessing L (2012) An analysis of design process models across disciplines. Proc Int Des Conf Des DS 70:171–180

Girard P, Doumeingts G (2004) Modelling the engineering design system to improve performance. Comput Ind Eng 46:43–67. https://doi.org/10.1016/j.cie.2003.09.008

Goevert K, Lindemann U (2018) Further development of an Agile technique toolbox for mechatronic product development. In: Proceedings of International Design Conference, DESIGN2018. Dubrovnik, Croatia, pp 2015–2026. https://doi.org/10.21278/idc.2018.0204

Graja I, Kallel S, Guermouche N et al (2018) A comprehensive survey on modeling of cyber-physical systems. Concurr Comput 32:1–18. https://doi.org/10.1002/cpe.4850

Guérineau J, Bricogne M, Rivest L, Durupt A (2022) Organizing the fragmented landscape of multidisciplinary product development: A mapping of approaches, processes, methods and tools from the scientific literature—Searchable cartographies. Zenodo. https://doi.org/10.5281/zenodo.6049224

Guérineau J, Rivest L, Bricogne M, et al (2018) Towards a design-method selection framework for multidisciplinary product development. In: Proceedings of International Design Conference, DESIGN2018. Dubrovnik, Croatia, pp 2879–2890. https://doi.org/10.21278/idc.2018.0431

Guérineau J, Rivest L, Bricogne M, Durupt A (2016) Agile and project-planned methods in multidisciplinary product design. In: Harik R, Rivest L, Bernard A, Eynard B, Bouras A (eds) Product lifecycle management for digital transformation of industries. PLM 2016. 13th IFIP Advances in Information and Communication Technology. Springer, Cham, pp 108–118. https://doi.org/10.1007/978-3-319-54660-5_11

Habchi G, Barthod C (2016) An overall methodology for reliability prediction of mechatronic systems design with industrial application. Reliab Eng Syst Saf 155:236–254. https://doi.org/10.1016/j.ress.2016.06.013

Habib T, Komoto H (2014) Comparative analysis of design concepts of mechatronics systems with a CAD tool for system architecting. Mechatronics 24:788–804. https://doi.org/10.1016/j.mechatronics.2014.03.003

Hamza G, Hammadi M, Barkallah M et al (2018) Conceptual design decision support of a mechatronic system using analytical approach with Modelica. Mech Ind 19:1–9. https://doi.org/10.1051/meca/2018019

Harashima F, Tomizuka M, Fukuda T (1996) Mechatronics—What is it, why, and how? IEEE/ASME Trans Mechatronics 1:1–4

Hehenberger P (2014) Perspectives on hierarchical modeling in mechatronic design. Adv Eng Informatics 28:188–197. https://doi.org/10.1016/j.aei.2014.06.005

Hehenberger P (2015) An approach to model-based parametric design of mechatronic systems. Comput Aided Des Appl 12:282–289. https://doi.org/10.1080/16864360.2014.981456

Hehenberger P, Bradley D (2016) Mechatronic Futures. Springer International Publishing

Hehenberger P, Poltschak F, Zeman K, Amrhein W (2010) Hierarchical design models in the mechatronic product development process of synchronous machines. Mechatronics 20:864–875. https://doi.org/10.1016/j.mechatronics.2010.04.003

Hehenberger P, Vogel-Heuser B, Bradley D et al (2016) Design, modelling, simulation and integration of cyber physical systems: methods and applications. Comput Ind 82:273–289. https://doi.org/10.1016/j.compind.2016.05.006

Herzog M, Bender B (2017) Competences for the development of smart products. In: Proc Int Conf Eng Des ICED17, pp 285–294

Highsmith J (2002) What is Agile software development? J Def Softw Eng 15:4–9. https://doi.org/10.1109/MC.2007.73

Hoxha B, Dokhanchi A, Fainekos G (2018) Mining parametric temporal logic properties in model-based design for cyber-physical systems. Int J Softw Tools Technol Transf 20:79–93. https://doi.org/10.1007/s10009-017-0447-4

Hu F, Lu Y, Vasilakos AV et al (2016) Robust cyber-physical systems: concept, models, and implementation. Futur Gener Comput Syst 56:449–475. https://doi.org/10.1016/j.future.2015.06.006

Hubka V, Eder WE (1988) Theory of technical systems: a total concept theory for engineering design, 2nd edn. Springer-Verlag, Berlin, Heidelberg

Hughes J, Cybenko G (2014) Three tenets for secure cyber-physical system design and assessment. Cyber Sens 2014:9097. https://doi.org/10.1117/12.2053933

Iansiti M, Lakhani KR (2014) Digital Ubiquity - How connections, sensors , and data, are revolutionizing business. Harv Bus Rev Nov:1–12

Ibarra D, Ganzarain J, Igartua JI (2018) Business model innovation through industry 4.0: a review. Procedia Manuf 22:4–10. https://doi.org/10.1016/j.promfg.2018.03.002

INCOSE (2015) Systems engineering handbook : a guide for system life cycle processes and activities, 4th edn. Wiley, San Diego, USA

Isasa JAE, Larsen PG, Hansen FO (2017) A holistic approach to energy-aware design of cyber-physical systems. Int J Embed Syst 9:283–295. https://doi.org/10.1504/IJES.2017.084682

Isermann R (1996a) On the design and control of mechatronic systems-a survey. IEEE Trans Ind Electron 43:4–15. https://doi.org/10.1109/41.481402

Isermann R (1996b) Modeling and design methodology for mechatronic systems. IEEE/ASME Trans Mechatronics 1:16–28. https://doi.org/10.1109/3516.491406

Isermann R (2002) Mechatronic design approach. In: Bishop RH (ed) The mechatronics handbook. CRC Press, pp 27–42

Isermann R (2006) Automotive Mechatronic Systems—General Developments and Examples (Mechatronische Systeme in Kraftfahrzeugen—Allgemeine Entwicklungen und Beispiele). Autom 54:419–429. https://doi.org/10.1524/auto.2006.54.9.419

Ishigooka T, Saissi H, Piper T et al (2017) Safety verification utilizing model-based development for safety critical cyber-physical systems. J Inf Process 25:797–810. https://doi.org/10.2197/ipsjjip.25.797

Jänsch J, Birkhofer H (2006) The development of the guideline VDI 2221—the change of direction. Int Des Conf - Des 2006:45–52

Janthong N, Brissaud D, Butdee S (2010) Combining axiomatic design and case-based reasoning in an innovative design methodology of mechatronics products. CIRP J Manuf Sci Technol 2:226–239. https://doi.org/10.1016/j.cirpj.2010.05.005

Jara AJ, Ladid L, Skarmeta A (2013) The internet of everything through IPv6: an analysis of challenges, solutions and opportunities. J Wirel Mob Netw Ubiquitous Comput Dependable Appl 4:97–118

Jensen JC, Chang DH, Lee EA (2011) A model-based design methodology for cyber-physical systems. IWCMC 2011—7th Int Wirel Commun Mob Comput Conf 1666–1671. https://doi.org/10.1109/IWCMC.2011.5982785

Jerke G, Lienig J, Freuer JB (2011) Constraint-Driven Design Methodology: A Path to Analog Design Automation. In: Graeb HE (ed) Analog Layout Synthesis. Springer US, Boston, MA, pp 269–297. https://doi.org/10.1007/978-1-4419-6932-3_7

Kagermann H, Wahlster W, Helbig J (2013) Recommendations for implementing the strategic initiative INDUSTRIE 4.0. Final Rep Ind 40 WG 82, acatech—National Academy of Science and Engineering.

Kang BG, Seo KM, Kim TG (2019) Model-based design of defense cyber-physical systems to analyze mission effectiveness and network performance. IEEE Access 7:42063–42080. https://doi.org/10.1109/ACCESS.2019.2907566

Kaul T, Meyer T, Sextro W (2017) Formulation of reliability-related objective functions for design of intelligent mechatronic systems. Proc Inst Mech Eng Part O J Risk Reliab 231:390–399. https://doi.org/10.1177/1748006X17709376

Kernschmidt K, Feldmann S, Vogel-Heuser B (2018) A model-based framework for increasing the interdisciplinary design of mechatronic production systems. J Eng Des 29:617–643. https://doi.org/10.1080/09544828.2018.1520205

Khaitan SK, McCalley JD (2015) Design techniques and applications of cyberphysical systems: a survey. IEEE Syst J 9:350–365. https://doi.org/10.1109/JSYST.2014.2322503

Kim JW, Sul SH, Choi JB (2018) Development of user customized smart keyboard using smart product design-finite element analysis process in the internet of things. ISA Trans 81:231–243. https://doi.org/10.1016/j.isatra.2018.05.010

Kiran CP, Clement S, Agrawal VP (2011) Design for X-abilities of a mechatronic system-a concurrent engineering and graph theory based approach. Concurr Eng Res Appl 19:55–70. https://doi.org/10.1177/1063293X11403509

Kleiner S, Kramer C (2013) Model Based Design with Systems Engineering Based on RFLP Using V6. In: Abramovici M, Stark R (eds) Smart Product Engineering, Proceedings of the 23rd CIRP Design Conference, Bochum, Germany, March 11th - 13th, 2013. Springer Heidelberg, Bochum, Germany, pp 93–102. https://doi.org/10.1007/978-3-642-30817-8_10

Knap L, Mączak J, Trojgo M (2018) Process-oriented approach to the design of cyber-physical systems. In: Awrejcewicz J. (ed) Dynamical systems in theoretical perspective. DSTA 2017. Springer Proceedings in Mathematics & Statistics. Springer, pp 193–204. https://doi.org/10.1007/978-3-319-96598-7_16

Komoto H, Tomiyama T (2010) A system architecting tool for mechatronic systems design. CIRP Ann - Manuf Technol 59:171–174. https://doi.org/10.1016/j.cirp.2010.03.104

Komoto H, Tomiyama T (2012) A framework for computer-aided conceptual design and its application to system architecting of mechatronics products. CAD Comput Aided Des 44:931–946. https://doi.org/10.1016/j.cad.2012.02.004

Lapusan C, Maties V, Balan R, Hancu O (2010) Rapid control prototyping in design process of mechatronic systems. Solid State Phenom 166–167:247–252. https://doi.org/10.4028/www.scientific.net/ssp.166-167.247

Larsen PG, Fitzgerald J, Woodcock J, et al (2016) Integrated tool chain for model-based design of Cyber-Physical Systems: The INTO-CPS project. In: 2016 2nd Int Work Model Anal Control Complex CPS, CPS Data 2016, pp 1–6. https://doi.org/10.1109/CPSData.2016.7496424

Lasi H, Fettke P, Kemper HG et al (2014) Industry 4.0. Bus Inf Syst Eng 6:239–242. https://doi.org/10.1007/s12599-014-0334-4

Lee EA, Seshia SA (2011) Introduction to Embedded Systems: A Cyber-physical Systems Approach, 1st edn. LeeSeshia.org

Lennon T (2008) Model-based design for mechatronic systems. In: EngineerIT, pp 16–20

Letia TS, Kilyen AO (2018) Using unified enhanced time petri net models for cyber-physical system development. IFAC-PapersOnLine 51:248–253. https://doi.org/10.1016/j.ifacol.2018.03.043

Li L (2017) China's manufacturing locus in 2025: With a comparison of "Made-in-China 2025" and "Industry 4.0." Technol Forecast Soc Change 135:66–74. https://doi.org/10.1016/j.techfore.2017.05.028

Li Q, Zhang WJ, Chen L (2001) Design for control-a concurrent engineering approach for mechatronic systems design. IEEE/ASME Trans Mechatronics 6:161–169. https://doi.org/10.1109/3516.928731

Li Y, Roy U, Saltz JS (2017) Modular Design of Data-Driven Analytics Models in Smart-Product Development. In: Proceedings of the ASME 2017 International Mechanical Engineering Congress and Exposition. Volume 11: Systems, Design, and Complexity. ASME. https://doi.org/10.1115/IMECE2017-71597

Liker JK, Morgan JM (2006) The toyota way in services: the case of lean product development. Acad Manag Perspect 20:5–20. https://doi.org/10.5465/AMP.2006.20591002

Lindemann U (2003) Methods are networks of methods. ICED 03:625–634

Lindemann U, Maurer M, Braun T (2009) Modeling the Multiple-Domain Matrix. In: Structural Complexity Management. Springer Heidelberg, Berlin, pp 67–78. https://doi.org/10.1007/978-3-540-87889-6_5

Liu J, Wang P, Lin J, Chu CH (2018) A Controller/Architecture Co-design Framework for Wireless Cyber-Physical Systems. In: Proc - 2017 IEEE Int Conf Internet Things, IEEE Green Comput Commun IEEE Cyber, Phys Soc Comput IEEE Smart Data, iThings-GreenCom-CPSCom-SmartData 2017, pp 76–84. https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData.2017.17

Liu J, Wu X (2019) Controller and architecture co-design of wireless cyber-physical systems. J Syst Archit 94:42–59. https://doi.org/10.1016/j.sysarc.2019.02.006

Liu J, Zhang L (2011) Aspect-oriented MDA development method for non-functional properties of cyber physical systems. In: Proc - 2nd Int Conf Netw Distrib Comput ICNDC 2011, pp 149–153. https://doi.org/10.1109/ICNDC.2011.76

Liu Y, Peng Y, Wang B et al (2017) Review on cyber-physical systems. IEEE/CAA J Autom Sin 4:27–40. https://doi.org/10.1109/JAS.2017.7510349

Luedeke TF, Köhler C, Conrad J, et al (2018) CPM / PDD in the context of Design Thinking and Agile Development of Cyber-Physical Systems. In: Ekströmer, Philip; Schütte S, Ölvander J (eds) DS 91: Proceedings of NordDesign 2018. Linköping University, Sweden, 14th - 17th August 2018, pp 1–10

Lukei M, Hassan B, Dumitrescu R et al (2016) Modular inspection equipment design for modular structured mechatronic products – model based systems engineering approach for an integrative product and production system development. Procedia Technol 26:455–464. https://doi.org/10.1016/j.protcy.2016.08.058

Mabrouk A, Penas O, Plateaux R, et al (2018) Integration of agility in a MBSE methodology for multidisciplinary systems design. In: Proc 4th IEEE Int Symp Syst Eng ISSE 2018, pp 1–5. https://doi.org/10.1109/SysEng.2018.8544425

Mahalik N, Xie C, Pu J, Moore PR (2006) Virtual distributed control systems: a components-based design method for mechatronic systems. Assem Autom 26:44–53. https://doi.org/10.1108/01445150610645657

Maier MW (1998) Architecting principles for systems-of-systems. Syst Eng 1:267–284. https://doi.org/10.1002/(SICI)1520-6858(1998)1:4%3c267::AID-SYS3%3e3.0.CO;2-D

Malik MA, Kayani SA (2008) Verifying experiment for automated design of mechatronic systems using Bond-Graph modelling and simulation and genetic programming. Int J Comput Appl Technol 32:173–180. https://doi.org/10.1504/ijcat.2008.020952

Malmquist D, Frede D, Wikander J (2014) Holistic design methodology for mechatronic systems. Proc Inst Mech Eng Part I J Syst Control Eng 228:741–757. https://doi.org/10.1177/0959651814527936

Marconnet B, Demoly F, Monticolo D, Gomes S (2017) An assembly oriented design and optimization approach for mechatronic system engineering. Int J Simul Multidiscip Des Optim 8:1–10. https://doi.org/10.1051/smdo/2016016

Mcharek M, Azib T, Hammadi M et al (2018) Knowledge sharing for mechatronic systems design and optimization. IFAC-PapersOnLine 51:1365–1370. https://doi.org/10.1016/j.ifacol.2018.08.338

Mcharek M, Hammadi M, Azib T et al (2019) Collaborative design process and product knowledge methodology for mechatronic systems. Comput Ind 105:213–228. https://doi.org/10.1016/j.compind.2018.12.008

Mellal MA, Adjerid S, Benazzouz D (2011) Modeling and simulation of mechatronic system to integrated design of supervision: using a bond graph approach. Appl Mech Mater 86:467–470. https://doi.org/10.4028/www.scientific.net/amm.86.467

Merlo C, Akle AA, Llaria A et al (2019) Proposal of a user-centred approach for CPS design: pillbox case study. IFAC-PapersOnLine 51:196–201. https://doi.org/10.1016/j.ifacol.2019.01.065

Merschak S, Hehenberger P (2019) Ecodesign Methods for Mechatronic Systems: A Literature Review and Classification. In: 2019 20th International Conference on Research and Education in Mechatronics (REM). IEEE, pp 1–8. https://doi.org/10.1109/REM.2019.8744105

Mhenni F, Choley JY, Penas O et al (2014) A SysML-based methodology for mechatronic systems architectural design. Adv Eng Informatics 28:218–231. https://doi.org/10.1016/j.aei.2014.03.006

Miranda J, Pérez-Rodríguez R, Borja V et al (2017) Sensing, smart and sustainable product development (S3product) reference framework. Int J Prod Res 57:4391–4412. https://doi.org/10.1080/00207543.2017.1401237

Moeuf A, Pellerin R, Lamouri S et al (2018) The industrial management of SMEs in the era of Industry 4.0. Int J Prod Res 56:1118–1136. https://doi.org/10.1080/00207543.2017.1372647

Mueller R, Thoring K (2012) Design Thinking vs. Lean Startup: A comparison of two user-driven innovation strategies. In: 2012 International Design Management Research Conference, pp 1–12

Nattermann R, Anderl R (2010) Approach for a Data-Management-System and a Proceeding-Model for the development of adaptronic systems. In: Proc ASME Int Mech Eng Congr Expo, pp 379–387. https://doi.org/10.1115/IMECE2010-37828

Nijssen EJ, Frambach RT (2000) Determinants of the adoption of new product development tools by industrial firms. Ind Mark Manag 29:121–131. https://doi.org/10.1016/S0019-8501(98)00043-1

Nunes ML, Pereira AC, Alves AC (2017) Smart products development approaches for Industry 4.0. Procedia Manuf 13:1215–1222. https://doi.org/10.1016/j.promfg.2017.09.035

Nuzzo P, Li J, Sangiovanni-Vincentelli AL et al (2019) Stochastic assume-guarantee contracts for cyber-physical system design. ACM Trans Embed Comput Syst 18:1–26. https://doi.org/10.1145/3243216

Nuzzo P, Sangiovanni-Vincentelli AL, Bresolin D et al (2015) A platform-based design methodology with contracts and related tools for the design of cyber-physical systems. Proc IEEE 103:2104–2132. https://doi.org/10.1109/JPROC.2015.2453253

Nyman J, Främling K (2008) Gathering Product Data from Smart Products. In: Cordeiro J, Filipe J (eds) 10th International Conference on Enterprise Information Systems. NSTICC-INST Syst Technologies Information Control & Communication, Barcelona, Spain, pp 252–257

O'Halloran BM, Papakonstantinou N, Van Bossuyt DL (2018) Assessing the Consequence of Cyber and Physical Malicious Attacks in Complex, Cyber-Physical Systems During Early System Design. In: 2018 IEEE 16th International Conference on Industrial Informatics (INDIN). IEEE, pp 733–740

Oestersötebier F, Just V, Trächtler A, et al (2012) Model-based design of mechatronic systems by means of semantic web ontologies and reusable solution elements. In: Proceedings of the ASME 2012 11th biennial conference on engineering systems design and analysis, volume 3: advanced composite materials and processing; robotics; information management and PLM; design engineering. ASME, pp 647–656. https://doi.org/10.1115/ESDA2012-82324

Osman K, Stamenkovic D, Lazarevic M (2013) Integration of system design and production processes in robust mechatronic product architectures development - extended M-FBFP framework. Hem Ind Ind 67:759–771. https://doi.org/10.2298/hemind121109003o

Paetzold K (2017) Product and Systems Engineering/CA* Tool Chains. In: Biffl S, Lüder A, Gerhard D (eds) Multi-Disciplinary Engineering for Cyber-Physical Production Systems. Springer International Publishing, pp 27–62. https://doi.org/10.1007/978-3-319-56345-9_2

Pahl G, Beitz W, Feldhusen J, Grote K-H (2007) Engineering design: a systematic approach. Springer, London. https://doi.org/10.1007/978-1-84628-319-2

Paluch S, Antons D, Brettel M et al (2019) Stage-gate and agile development in the digital age: promises, perils, and boundary conditions. J Bus Res 110:495–501. https://doi.org/10.1016/j.jbusres.2019.01.063

Pannaga N, Ganesh N, Gupta R (2013) Mechatronics—an introduction to mechatronics. Int J Eng 2:128–134. 0849363586

Penas O, Plateaux R, Choley J-Y, et al (2010) Conception mécatronique - Vers un processus continu de conception mécatronique intégrée. Tech l'Ingénieur. https://doi.org/10.51257/a-v1-bm8020

Penas O, Plateaux R, Patalano S, Hammadi M (2017) Multi-scale approach from mechatronic to cyber-physical systems for the design of manufacturing systems. Comput Ind 86:52–69. https://doi.org/10.1016/j.compind.2016.12.001

Pérez-Rodríguez R, Molina A, Miranda J, et al (2018) The requirements specification of mechatronic products based-on the Integrated product, process and manufacturing development model. Ing Mec sep-dic:145–152. 133573268

Pietrusewicz K (2019) Metamodelling for design of mechatronic and cyber-physical systems. Appl Sci 376:1–40. https://doi.org/10.3390/app9030376

Plateaux R, Penas O, Choley J-Y et al (2010) Integrated design methodology of a mechatronic system. Mécanique Ind 11:401–406. https://doi.org/10.1051/meca/2010052

Plateaux R, Penas O, Mule S, et al (2020) SCRUM++ Framework concepts. In: 2020 IEEE International Symposium on Systems Engineering (ISSE). IEEE, pp 1–8. https://doi.org/10.1109/ISSE49799.2020.9272233

Porter ME, Heppelmann JE (2014) How smart, Connected Products Are Transforming Competition. Harv Bus Rev Oct:97–114. https://doi.org/10.1017/CBO9781107415324.004

Rothful R, Lasa M, Heinkel HM, Tirgari P (2006) Systems engineering in the design of mechatronic systems. Int J Veh Des 28:1–17. https://doi.org/10.1504/ijvd.2002.001976

Raghav G, Gopalswamy S (2009) Architecture driven development for cyber physical systems. SAE Int J Aerosp 3:95–100. https://doi.org/10.4271/2009-01-3263

Rahimi N, Ibarra M (2014) A Review of Multiple User Center Design Methods for New Product Development in Smart and Connected Health Applications. In: 2014 Portl Int Conf Manag Eng Technol, pp 3498–3510

Rajkumar R (2012) A Cyber-Physical Future Proc IEEE 100:1309–1312. https://doi.org/10.1109/JPROC.2012.2189915

Rajkumar R, Lee I, Sha L, Stankovic J (2010) Cyber-physical systems: the next computing revolution. In: Proc 47th Des Autom Conf—DAC '10, pp 731–736. https://doi.org/10.1145/1837274.1837461

Rauch E, Dallasega P, Matt DT (2016) The way from lean product development (LPD) to smart product development (SPD). Procedia CIRP 50:26–31. https://doi.org/10.1016/j.procir.2016.05.081

Rauniyar D, Tanik U (2010) Integrated product development approach for cyber-physical system utilizing standardized modelling languages and methodologies. Trans SDPS 14:1–11

Renner FM, Hoffmann KJ, Markert R, Glesner M (2000) Design methodology of application specific integrated circuits for mechatronic systems. Microprocess Microsyst 24:95–103. https://doi.org/10.1016/S0141-9331(99)00071-X

Riel A, Kreiner C, Messnarz R, Much A (2018) An architectural approach to the integration of safety and security requirements in smart products and systems design. CIRP Ann 67:173–176. https://doi.org/10.1016/j.cirp.2018.04.022

Ries E (2011) The Lean Startup. Crown Business, New York, USA

Rijsdijk SA, Hultink EJ (2009) How today's consumers perceive tomorrow's smart products. J Prod Innov Manag 26:24–42. https://doi.org/10.1111/j.1540-5885.2009.00332.x

Roblek V, Meško M, Krapež A (2016) A Complex View of Industry 4.0. SAGE Open 6:1–11. https://doi.org/10.1177/2158244016653987

Roozenburg NFM, Eekels J (1995) Product design : fundamentals and methods. Wiley

Royce WW (1970) Managing the development of large software systems. In: IEEE Wescon, pp 328–338

Rúbio EM, Dionísio RP, Torres PMB (2019) Industrial IoT devices and cyber-physical production systems: review and use case. In: Machado J, Soares F, Veiga G (eds) Innovation, engineering and entrepreneurship. HELIX 2018. pp 292–298

Rui-Qin L, Hui-Jun Z (2005) A new symbolic representation method to support conceptual design of mechatronic system. Int J Adv Manuf Technol 25:619–627. https://doi.org/10.1007/s00170-004-1904-2

Rüßmann M, Lorenz M, Gerbert P, et al (2015) Industry 4.0 The Future of Productivity and Growth in Manufacturing Industries, Boston Consulting Group report.

Säde S (1999) Representations of smart product concepts in user interface design. In: Jordan PW (ed) Green WS. Current Practice and Future Trends. CRC Press, Human factors in product design, pp 64–72

Sadlauer A, Hehenberger P (2017) Using design languages in model-based mechatronic system design processes. Int J Agil Syst Manag 10:73–91. https://doi.org/10.1504/ijasm.2017.10003875

Salman T, Jain R (2016) Networking protocols and standards for Internet of Things. In: Geng H (ed) Internet of things and data analytics handbook. Wiley, Hoboken, NJ, USA, pp 215–238

Salminen V, Verho A (1992) Systematic and innovative design of a mechatronic product. Mechatronics 2:257–275. https://doi.org/10.1016/0957-4158(92)90020-O

Sangiovanni-Vincentelli A, Carloni L, De Bernardinis F, Sgroi M (2004) Benefits and challenges for platform-based design. In: DAC '04: Proceedings of the 41st annual Design Automation Conference, pp 409–414. https://doi.org/10.1145/996566.996684

Sangiovanni-Vincentelli A, Damm W, Passerone R (2012) Taming Dr. frankenstein: contract-based design for cyber-physical systems*. Eur J Control 18:217–238. https://doi.org/10.3166/ejc.18.217-238

Sangregorio P, Cologni AL, Piccinini A et al (2015) A method for automation software design of mechatronic systems in manufacturing. IFAC-PapersOnLine 28:936–941. https://doi.org/10.1016/j.ifacol.2015.06.203

Santos C, Mehrsai A, Barros AC et al (2017) Towards industry 4.0: an overview of European strategic roadmaps. Procedia Manuf 13:972–979. https://doi.org/10.1016/j.promfg.2017.09.093

Schmüdderrich T, Trächtler A, Brökelmann J, Gausemeier J (2013) Procedural Model for the Virtual Commissioning on the Basis of Model-Based Design. In: Abramovici M, Stark R (eds) Smart Product Engineering (23rd CIRP Design Conference). Springer Heidelberg, Berlin, pp 23–32. https://doi.org/10.1007/978-3-642-30817-8

Schuh G, Rudolf S, Breunig S (2016a) Modular platform design for mechatronic systems using axiomatic design and mechatronic function modules. Procedia CIRP 50:701–706. https://doi.org/10.1016/j.procir.2016.05.035

Schuh G, Rudolf S, Riesener M (2016b) Design for industrie 4.0. In: Proc Int Des Conf DESIGN2016b DS 84, pp 1387–1396. https://doi.org/10.1109/HICSS.2016b.488

Schwaber K (1997) SCRUM Development Process. In: Sutherland J, Casanave C, Miller J, Patel P, Hollowell G (eds) Business Object Design and Implementation. Springer, London, pp 117–134. https://doi.org/10.1007/978-1-4471-0947-1_11

Schweiger W, Levichev D, Schön A, Meerkamm H (1999) Digital mock-up in the early stages of mechatronic product design. J Eng Des 10:235–246. https://doi.org/10.1080/095448299261308

Seshia SA, Hu S, Li W, Zhu Q (2017) Design automation of cyber-physical systems: challenges, advances, and opportunities. IEEE Trans Comput Des Integr Circuits Syst 36:1421–1434. https://doi.org/10.1109/TCAD.2016.2633961

Sierla S, Tumer I, Papakonstantinou N et al (2012) Early integration of safety to the mechatronic system design process by the functional failure identification and propagation framework. Mechatronics 22:137–151. https://doi.org/10.1016/j.mechatronics.2012.01.003

Sini J, Violante M, Dessi R (2018) Computer-Aided Design of Multi-Agent Cyber-Physical Systems. In: 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, pp 677–684. https://doi.org/10.1109/ETFA.2018.8502448

Slomka F, Kollmann S, Moser S, Kempf K (2011) A multidisciplinary design methodology for cyber-physical systems. CEUR Workshop Proc 795:23–37. https://doi.org/10.1109/IWCMC.2011.5982785

Sommerville I (2010) Software Engineering, 9th edn. Pearson

Song Y, Park J, Lee KC, Lee S (2019) Reducing the design complexity of automated vehicle electrical and electronic systems using a cyber-physical system concept. Int J Control Autom Syst 17:500–508. https://doi.org/10.1007/s12555-018-0246-z

Stelzmann E (2012) Contextualizing agile systems engineering. IEEE Aerosp Electron Syst Mag 27:17–22. https://doi.org/10.1109/MAES.2012.6226690

Stetter R, Lindemann U (2005) The transfer of methods into industry. In: Clarkson J, Eckert C (eds) Design process improvement. Springer, London, pp 436–459

Suh NP (1998) Axiomatic design theory for systems. Res Eng Des 10:189–209. https://doi.org/10.1007/s001639870001

Sünnetcioglu A, Brandenburg E, Rothenburg U, Stark R (2016) ModelTracer: user-friendly traceability for the development of mechatronic products. Procedia Technol 26:365–373. https://doi.org/10.1016/j.protcy.2016.08.047

Sztipanovits J, Bapty T, Koutsoukos X et al (2018) Model and tool integration platforms for cyber-physical system design. Proc IEEE 106:1501–1526. https://doi.org/10.1109/JPROC.2018.2838530

Tariq MU, Florence J, Wolf M (2018) Improving the safety and security of wide-area cyber-physical systems through a resource-aware, service-oriented development methodology. Proc IEEE 106:144–159. https://doi.org/10.1109/JPROC.2017.2744645

Thebault P (2013) La conception à l'ère de l'Internet des Objets : modèles et principes pour le design de produits aux fonctions augmentées par des applications. Dissertation, École Nationale Supérieure d'Arts et Métiers, ENSAM

Thoring K, Müller RM (2011) Understanding Design Thinking: a process model based on method engineering. Proceedings of E&PDE 2011, the 13th International Conference on Engineering and Product Design Education, 69:493–498.

Tomiyama T, Gu P, Jin Y et al (2009) Design methodologies: Industrial and educational applications. CIRP Ann—Manuf Technol 58:543–565. https://doi.org/10.1016/j.cirp.2009.09.003

Tomiyama T, Lutters E, Stark R, Abramovici M (2019) Development capabilities for smart products. CIRP Ann 68:727–750. https://doi.org/10.1016/j.cirp.2019.05.010

Tomizuka M (2006) Sensing rich design of drive trains for mechatronic systems. Smart Struct Mater 2006 Sensors Smart Struct Technol Civil. Mech Aerosp Syst 6174:1–7. https://doi.org/10.1117/12.659962

Turki S, Soriano T, Sghaier A (2005) A SysML profile for mechatronics integrating bond graphs. 9th WSEAS international conference circuits, communications, computers (CSCC'05). Greece, Athens, pp 1–6

Ulrich KT, Eppinger SD (2004) Product design and development, 3rd edn. McGraw-Hill, New York, USA

Ulrich KT, Eppinger SD (2016) Product design and development, 6th edn. McGraw-Hill, New York, USA

Umeda Y, Ishii M, Yoshioka M et al (1996) Supporting conceptual design based on the function-behavior-state modeler. Artif Intell Eng Des Anal Manuf 10:275–288. https://doi.org/10.1017/s0890060400001621

Vachtsevanos G, Lee B, Oh S, Balchanos M (2018) Resilient design and operation of cyber physical systems with emphasis on unmanned autonomous systems. J Intell Robot Syst Theory Appl 91:59–83. https://doi.org/10.1007/s10846-018-0881-x

Vajna S, Clement S, Jordan A, Bercsey T (2005) The autogenetic design theory: an evolutionary view of the design process. J Eng Des 16:423–440. https://doi.org/10.1080/09544820500267781

Valasek M (2016) Highly efficient models and simulations—the basis of design of mechatronic systems. Metallurgy 49:127–131

Van Beek TJ, Erden MS, Tomiyama T (2010) Modular design of mechatronic systems with function modeling. Mechatronics 20:850–863. https://doi.org/10.1016/j.mechatronics.2010.02.002

Vasić VS, Lazarević MP (2008) Standard industrial guideline for mechatronic product design. FME Trans 36:103–108

VDI-Fachbereich Produktentwicklung und Mechatronik (2004) VDI-Guideline 2206 : design methodology for mechatronic system

Veitl A, Gordon T, Van de Sand A et al (2000) Methodologies for coupling simulation models and codes in mechatronic system analysis and design. Dyn Veh Roads Tracks 33:231–243. https://doi.org/10.1080/00423114.1999.12063084

Violante MG, Vezzetti E (2017) Kano qualitative vs quantitative approaches: an assessment framework for products attributes analysis. Comput Ind 86:15–25. https://doi.org/10.1016/j.compind.2016.12.007

Wagner S (2014) Scrum for cyber-physical systems: a process proposal. In: Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering—RCoSE 2014. ACM Press, New York, USA, pp 51–56. https://doi.org/10.1145/2593812.2593819

Wan J, Canedo A, Al Faruque MA (2017) Functional model-based design methodology for automotive cyber-physical systems. IEEE Syst J 11:2028–2039. https://doi.org/10.1109/JSYST.2014.2387487

Wang J, Fan Z, Terpenny JP, Goodman ED (2005) Knowledge interaction with genetic programming in mechatronic systems design using bond graphs. IEEE Trans Syst Man Cybern Part C Appl Rev 35:172–182. https://doi.org/10.1109/TSMCC.2004.841915

Warniez A, Penas O, Soriano T (2012) About metrics for integrated mechatronic system design. In: 2012 9th France-Japan & 7th Europe-Asia congress on mechatronics (MECATRONICS) / 13th Int'l Workshop on Research and Education in Mechatronics (REM). IEEE, pp 450–457

Wiener N (1965) Cybernetics Or Control and Communication in the Animal and the Machine, 2nd edn. MIT Press, Cambridge, Massachusetts

Woo H, Yi J, Browne JC, et al (2008) Design and development methodology for resilient cyber-physical systems. In: Proc - Int Conf Distrib Comput Syst, pp 525–528. https://doi.org/10.1109/ICDCS.Workshops.2008.62

Wrobel J, Jezewski J, Horoba K et al (2015) Medical cyber-physical system for home telecare of high-risk pregnancy: design challenges and requirements. J Med Imaging Heal Informatics 5:1295–1301. https://doi.org/10.1166/jmihi.2015.1532

Wu FL, He HB, Deng YM (2012) Applying bond graphs for design representation of mechatronic products. Appl Mech Mater 201–202:537–540. https://doi.org/10.4028/www.scientific.net/amm.201-202.537

Wu JC, Leu MC, Liu XF (2009) A hierarchical object-oriented functional modeling framework for co-design of mechatronic products. Concurr Eng Res Appl 17:245–256. https://doi.org/10.1177/1063293X09352120

Xu Y, Zou H (2007) Design principles for mechatronic systems based on information content. Proc Inst Mech Eng Part B J Eng Manuf 221:1245–1254. https://doi.org/10.1243/09544054JEM699

Yuan H, Xia Y (2018) Resilient strategy design for cyber-physical system under DoS attack over a multi-channel framework. Inf Sci (ny) 454–455:312–327. https://doi.org/10.1016/j.ins.2018.04.082

Yuan W, Liu Y, Zhao J, Wang H (2016) Pattern-based integration of system optimization in mechatronic system design. Adv Eng Softw 98:23–37. https://doi.org/10.1016/j.advengsoft.2016.03.005

Zhang L (2011a) Aspect-oriented MDA approach for non-functional properties of distributed cyber physical systems. In: Proc - 2011a 10th Int Symp Distrib Comput Appl to Business, Eng Sci DCABES 2011a, pp 284–288. https://doi.org/10.1109/DCABES.2011a a.26

Zhang L (2011b) MDA approach for non-functional characteristics of cyber physical systems based on aspect-oriented method. Commun Comput Inf Sci 215:323–328. https://doi.org/10.1007/978-3-642-23324-1_52

Zheng C, Bricogne M, Le Duigou J, Eynard B (2014) Survey on mechatronic engineering: a focus on design methods and product models. Adv Eng Informatics 28:241–257. https://doi.org/10.1016/j.aei.2014.05.003

Zheng C, Hehenberger P, Le Duigou J et al (2017) Multidisciplinary design methodology for mechatronic systems based on interface model. Res Eng Des 28:333–356. https://doi.org/10.1007/s00163-016-0243-2

Zheng C, Le Duigou J, Bricogne M, et al (2016) Multidisciplinary Interface Modelling: A Case Study on the Design of 3D Measurement System. In: Bouras A, Eynard B, Foufou S, Thoben KD (eds) Product Lifecycle Management in the Era of Internet of Things, PLM 2015, IFIP Advances in Information and Communication Technology. Springer, Cham, pp 856–866. https://doi.org/10.1007/978-3-319-33111-9_78

Zheng C, Qin X, Eynard B et al (2019) Interface model-based configuration design of mechatronic systems for industrial manufacturing applications. Robot Comput Integr Manuf 59:373–384. https://doi.org/10.1016/j.rcim.2019.05.011

Zheng P, Xu X, Chen C-H (2018) A data-driven cyber-physical approach for personalised smart, connected product co-development in a cloud-based environment. J Intell Manuf 31:3–18. https://doi.org/10.1007/s10845-018-1430-y

Zhu Q, Sangiovanni-Vincentelli A (2018) Codesign Methodologies and Tools for Cyber-Physical Systems. Proc IEEE 106:1484–1500. https://doi.org/10.1109/JPROC.2018.2864271

Zou J, Du Q (2013) A functional reasoning cube model for conceptual design of mechatronic systems. Stroj Vestnik/journal Mech Eng 59:323–332. https://doi.org/10.5545/sv-jme.2012.757