Title:

Future Trends in CAD – from the Perspective of Automotive Industry

Authors:
Mario Hirz, mario.hirz@tugraz.at, Graz University of Technology
Patrick Rossbacher, patrick.rossbacher@tugraz.at, Graz University of Technology
Jana Gavačová, xgavacova@stuba.sk, Slovak University of Technology in Bratislava

Keywords:
future CAD, visualization technology, adaptable models, generic programming, development processes

DOI: 10.14733/cadconfP.2016.251-256

Introduction:

Computer-aided design (CAD) plays a central role in automotive development. 3D-CAD models not only provide geometry information, but also serve as a basis for the configuration of modules and systems, as well as for different simulation and verification processes. Product assemblies within CAD-platforms include structure-related information, are used for digital mock-up (DMU) investigations and provide lots of data for production and manufacturing engineering. Since first CAD systems have been applied in automotive industry in the 1980s, design software has been developed steadily so that today's CAD programs are able to provide far-reaching possibilities for integrated product development.

The present work includes an introduction and evaluation of existing and upcoming technologies for their use in future CAD programs and discusses different scenarios of implementation into automotive development. Vehicle architects, design engineers, project managers, IT-administrators for CAD, stylists as well as styling engineers have contributed to a survey at automotive manufacturer and supplier. The results of this survey are brought into this publication to discuss the future role of CAD in the development of new cars and to define requirements for the next generation of CAD software.

Integration of CAD into automotive development

Through the middle 1980s, automotive development processes were characterized by a high degree of hardware and prototype-based optimization cycles. At that time, full-vehicle development had a duration of about 6 years and included three prototype phases. The integration of virtual, computer-aided methods into automotive development led to significantly revised processes. In the middle of the 1980s, computer-aided design and simulation methods began taking over engineering tasks that had formerly been done via hardware based development. In the closing years of the twentieth century, integrated CAD/CAE processes enabled network-based development in automotive engineering. Virtual prototypes were generated, which replaced at least one physical prototype generation. The application of virtual engineering in full-vehicle development fostered worldwide collaboration by bringing together partners and markets from different countries and regions. Today, a full-vehicle development project takes less than 3 years, and the trend is moving towards an additional decrease [7]. In state of the art automotive development, nearly all process phases are related to CAD data or processes; most of the styling, design and simulation works are done in CAD or based on CAD data. Even disciplines that do not include CAD work show a certain involvement of CAD data: for evaluation purposes, prototyping and data exchange.

First commercial CAD programs came up in the 1970s and provided functions for 2D-drawings and archival. 3D-CAD design changed the applied design methods significantly in the 1980s. Most importantly, the introduction of 3D surface and solid models resulted in an evolution of design methods from static, two-dimensional drawings in several views and sections to a dynamic creation of three-dimensional virtual geometric product models. Besides a detailed and near-real-life representation of product geometry, these models included a variety of additional information and characteristics.

With the help of 3D design, it was possible to integrate production-related knowledge or assembly-related information into the models [7]. Direct data exchange between design and simulation started in the 1990s through the use of standardized neutral data exchange formats. In this way it was possible to use geometry data defined in design software packages for the definition of product geometries in simulation processes. The parameterization of geometry data represented a significant evolutionary step in 3D-CAD processes by separation of geometry administration and its controlling parameters. Parameter-based control of geometrical model characteristics opened up a wide field for creation of problem-specific design applications. Parameterized CAD programs offered additional functionalities, such as data interfaces, the integration of catalogue and knowledge ware functions, and the possibility of macro-based procedures. All of these functionalities characterize state of-the-art CAD packages, which have come into use in automotive development.

Future CAD systems: easy to handle design software or integrated development platform?

One trend in software industry is going into the direction of integrated packages, which combine parametric design programs and simulation software in the same environment. Although this strategy reduces data interface losses, it has been criticized for resulting functional reduction in directly compatible program platforms. Nevertheless, future intelligent geometry data exchange formats, which will be able to handle both geometry data and enhanced product information, are able to increase the efficiency of virtual product development processes significantly. Virtual development will be extended to the entire range of product generation, starting from concept phase, continuing in different development steps (including manufacturing and production engineering), supporting sales and aftermarket, and ending in the organization of disposal processes. This requires a further integration of product characteristics into 3D models, so that they are able to behave as in reality.

This time, there are several integrated development platforms available, but they suffer from high complexity, leading to disadvantages in operation efficiency and user handling. Overloaded with interlinked software modules for design, digital mock-up, simulation and data management, these platforms have grown to huge software giants, which offer more functionalities than ever before. Future development platforms have to provide a smarter integration of different disciplines, so that they are more efficient and easier to handle. Besides that, the licensing strategies have to change from rigid, module-oriented systems to flexible accounting, which is able to consider individual software usage. Alternatives to traditional CAD software will come up with an increasing share of open-source CAD systems [9], [1]. Today's open source CAD, like BRL-CAD [2], are not so widespread yet, but there is a big demand for simple, flexible CAD programs especially in small- and mid-size supplier companies. Similar to Linux [10], OpenOffice [16] or Firebird [6], this type of programs will be based on open-source platforms like the Open CASCADE engine [15], and be able to provide sufficient functionalities for product and systems design [19]. One big challenge for the development of these software packages represents the possibility of integrating created data into existing data management structures of big (car) manufacturers.

In general, collaborating CAD systems and environments play an increasingly important role in the worldwide acting automotive industry. To face this challenge, future CAD systems will provide functionalities of cloud-based development that enables every engineer on a design team simultaneously work together using a web browser, phone or tablet, as it is provided by some programs today, e.g. [14]. In this way, compatibility between different CAD platforms and data management systems represents itself a pre-condition for a high level of interoperability [3].

Visualization and user interfaces:

State-of-the-art CAD systems are controlled by keyboard in combination with operators like mouse, space mouse or pads. With introduction of virtual and augmented reality technologies, additional functionalities of virtual model visualization and handling came up during the past decade. This time, new technologies are approaching, which provide an integration of 3D models, virtual and real environment, e.g. the Microsoft HoloLens [13]. Most of these applications stem from gaming industry, but are now on a higher technological level so that they can be used in industrial development processes, too. Holographic animation of virtual models will be used for styling, design and DMU applications including manipulation and kinematics simulation. Especially the integration of DMU and real scenarios enables real-time development and evaluation in a collaborative way, independently from the actual geographic location of involved engineers. These might be the biggest benefits of holographic technologies: styling evaluation, packaging and assembly investigations. From engineering-oriented point of view, holographic technologies are seen more critical yet. Component and module design and engineering come with a more difficult integration of descriptive technical information into visualization technologies. In this way, future CAD systems will work within virtual environment to gather boundary conditions for subsequent technical development of components and systems. CAD models will be integrated in holographic DMUs to be investigated, and information will be fed back into design processes. One big future challenge in the development of comprehensive virtual environment for product design includes the integration of the mentioned working steps into one efficient workflow.

3D printing and adaptable hardware models:

3D printing has been established as a capable way to produce hardware models out of virtual 3D-CAD geometry data. During the past years, new technologies provided the possibility to use new materials with specific characteristics, e.g. different types of plastics, metal and even textile and ceramics. This allows in-time generation of hardware models during product development, with characteristics near to those out of serial production. Due to enhanced production capabilities, 3D printing is actually even able to supply mass production processes. The reduction of costs makes 3D printing interesting for a deep integration into future product development, with a certain share of virtual modelling and hardware-based innovation finding.

To bring hardware and virtual modelling closer into one development process, new “intelligent” materials are topic of research. Exemplary, Dynamic Physical Rendering (DPR) is a collaborative research project between Carnegie Mellon University in Pittsburgh and Intel [4]. Target is the development of flexible material based on nanotechnology that can be controlled in two ways: by geometry information from virtual 3D-CAD models and by manual manipulation. Once being modified by hand, the geometrical modifications are transferred back into the CAD system automatically. The technology is based on millions of so-called catoms, which represent tiny particles within a type of plasticine. A 3D printing process forms a hardware model out of these catoms, and afterwards the model can be reshaped with fingers or tools. All geometrical modifications are handed over to the CAD system and modify the behind lying virtual model. Because of high computational effort to manage shaping of millions of catoms into dynamic 3D forms, new approaches for reliable and robust software programs are required.

The DPR concept, also known as “Claytronics”, opens a wide field of application in future product design. Once available for industrial applications, this technology will directly support styling and shaping work because of the possibility to integrate intuitive-oriented artwork and technological-oriented engineering tasks. In addition, resource-consuming loops of clay modelling, 3D scanning and shape optimization of virtual models will be reduced, and focusing on creative and productive works will be assisted. The integration of DPR into technological-oriented engineering processes is a bigger challenge. 3D-CAD modelling of components and modules requires the integration of numerous technological, functional and production-related aspects into product creation. Time will tell if and how this new technology will be integrated into technical product design and evaluation processes.

CAD specialization and generic programming:

State-of-the-art CAD systems provide functionalities for knowledge-based engineering to support the integration of template models, automated computation routines and even programs into CAD models. These possibilities offer a good basis for shortening development time and increasing product quality at the same time [17]. Although opportunities of customizing are given in modern CAD systems so far, both functionalities and handling show drawbacks in comparison with commercial available scientific programs, like MATLAB [11]. The applied programming languages, e.g. VBScript [12] are not up to date for efficient programming within CAD environment, because they show disadvantages in deployment and portability as well as in complex programming, e.g. ontological and generic functionalities, working with classes. In addition, they offer just restricted support of enhanced operations and collaboration with other software packages.

Future CAD platforms will provide enhanced object-oriented programming with integrated functions and operations for the efficient creation of problem-specific solutions [8]. Design processes will increasingly change from purely geometry creation to integrated development cycles including efficient layout and simulation. This will be enabled not only by use of integrated standard simulation packages, but also by specialized CAD automatisms, which support design engineers in their work without overloading them with management of confusing software packages. In addition, future CAD systems will be able to learn from previous development and to support the engineers with optimization processes in the background. In this way, the engineers will be supplied with suggestions for their actual design problem, which are based on knowledge delivered by automated algorithms and databases.

The development of complex components, modules and systems, as they occur in automotive industry, often includes an optimization of numerous, partially conflicting aspects. Higher design engineering quality leads to lower number of modification cycles, more efficient development processes and better product quality. Quality of results depends on the experience of involved design engineers, but the support of applied software also plays an important role. Facing this challenge, future CAD systems will increasingly provide optimization functionalities, which are able to assist engineers when starting from initial product creation phases. One topic of research includes the implementation of genetic programming into CAD environment by use of evolutionary algorithms to optimize populations of designs according to certain criteria. Originating from the development of computer programs, genetic programming offers new approaches for the design of complex products that are composed of numerous components.

Upcoming technologies require new development processes:

Increasing integration of electrics and electronics (E/E) systems into cars is motivated by aspects of comfort and communication, but also by continuously strengthened legislative boundary conditions for exhaust emissions and vehicle safety. In a modern midsize car, the share of averaged value creation of E/E reaches more than 30%, and this value will further increase during upcoming years. However, E/E research and development has overtaken traditional mechanical-oriented development tasks and it will become more important during the next years. The development processes did not follow the new requirements so far, and applied development software lacks in this context, too. Future CAD systems have to face these new product characteristics by provision of enhanced development methods for mechatronics systems, which are able to support simultaneous development of mechanical, electrical and electronics systems including software. Especially software development follows different rules than hardware development, because of differing behavior in view of complexity management, occurring errors and failure, durability and lifecycle. In this way, future CAD systems have to provide functionalities not only to implement e.g. an electric motor or a camera system into a DMU, but also features to consider functional aspects of implemented modules. For the applied CAD platforms, this requires interfaces to software development programs and evaluation tools for testing of complete mechatronics systems.

Figure 1 shows a typically development process of automotive mechatronics systems according to the so-called V-model [5]. The V-model represents a special case of a waterfall model that describes sequential steps of product development [18]. The process begins at the top end of the left branch with product specifications that result from a list of requirements. The entire left branch focusses on product design and is divided into a sequential chronology of increasing levels of detail. The System level includes full-vehicle related development, e.g. vehicle architecture, packaging, and of course styling. After having defined main characteristics on full-vehicle level, the Module level includes a breakdown of complex systems into several modules, e.g. vehicle body, drivetrain, chassis, comfort and driving assistance modules. Finally, modules are divided into their components, which are developed in the Component level. Cross-domain implementation is performed at the bottom level of the V-model in the course of component integration. Today, this is mainly done by product-oriented processes, which focus on product characteristics and functionalities, but not by effective process integration.

The right branch of the V-model includes prototyping, testing and optimization at Component, Module and System level. After being tested, components are built together to modules, which are integrated and tested according to their specific functionalities. In the final System level at the top end of the right branch, all elements are assembled to a full-vehicle prototype and tested for product confirmation according to initially defined specifications. Typically for development according to the V-model is a close interaction of design and testing. In this way, data and information exchange between product design (left branch) and integration & testing (right branch) support efficient improvements and optimization. In case of highly complex products, e.g. cars, the development process is run through several times, especially on Module and Component level. Both duration and complexity of these development cycles differ significantly in the three domains, which leads to varying levels of maturities in mechanic, hardware and software development.

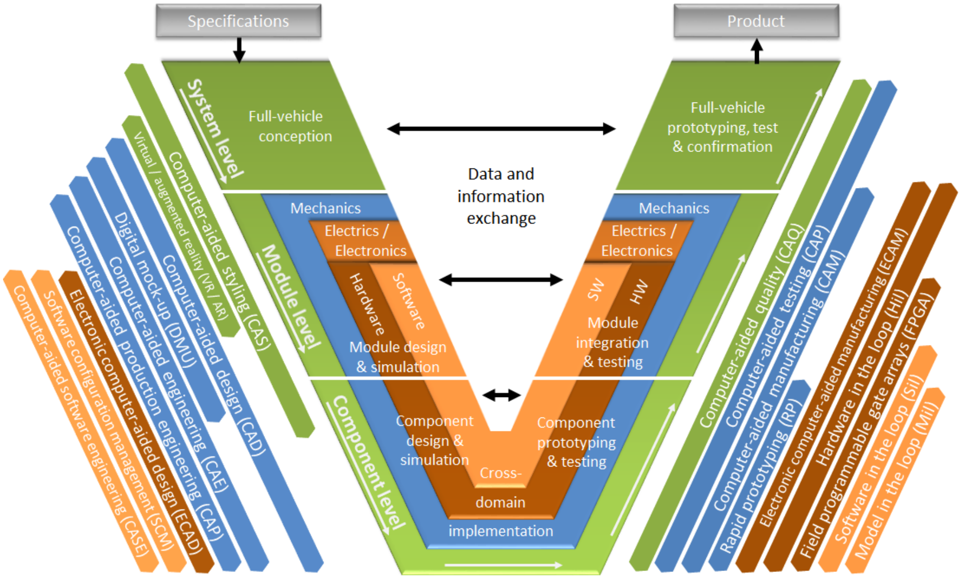
Specifications
Product
System level
Full-vehicle conception
Full-vehicle prototyping, test & confirmation
Data and information exchange
Module level
Mechanics
Electrics / Electronics
Hardware
Software
Module design & simulation
Mechanics
Electrics / Electronics
SW
HW
Module integration & testing
Component level
Component design & simulation
Component prototyping & testing
Cross-domain implementation
Computer-aided styling (CAS)
Virtual / augmented reality (VR / AR)
Computer-aided design (CAD)
Digital mock-up (DMU)
Computer-aided engineering (CAE)
Computer-aided production engineering (CAP)
Electronic computer-aided design (ECAD)
Software configuration management (SCM)
Computer-aided software engineering (CASE)
Computer-aided quality (CAQ)
Computer-aided testing (CAP)
Computer-aided manufacturing (CAM)
Rapid prototyping (RP)
Electronic computer-aided manufacturing (ECAM)
Hardware in the loop (HiL)
Field programmable gate arrays (FPGA)
Software in the loop (SiL)
Model in the loop (MiL)

Fig. 1: Different domains and computer-aided applications in an automotive development process according to the V-model.

Figure 1 also points out different domains that occur in the development of mechatronics systems: mechanics (blue), and electrics/electronics (E/E) (orange). E/E is divided into hardware (dark orange) and software development (light orange). Bars next to the left and right branch indicate the application of different computer-aided tools during the development process. The colors of the bars correspond with the development domains mechanics, hardware and software. “Traditional” development disciplines in the mechanical domain are dominated by CAD-related applications, e.g. CAS, DMU, CAP and different types of CAE, e.g. finite-elements simulation (FEM), multi-body simulation (MBS) and computational fluid dynamics simulation (CFD). Due to the central role of CAD-based product models, which contain geometrical, functional, structural and manufacturing-related information, several disciplines at the right branch of the V-model are also related to CAD, e.g. CAP, CAM, RP and many aspects of CAQ. In today’s development processes, E/E development is quite decoupled from mechanics development. There is a certain interaction in the areas of ECAD and ECAM, but not in software development, integration and testing.

In general, the V-model represents itself as a well-established approach for the development of mechatronics systems, but with rising complexity and increasing share of E/E, it shows disadvantages in terms of efficient cross-domain integration. To face these challenges, future development processes will provide functionalities for effective integration of geometric, structural and functional design, not restricted to mechanical systems, but also including hard- and software. One key of success lies in the introduction of flexible, interdisciplinary processes, which are able to consider different domain-specific methodological, functional and development-cycle-time related characteristics. In addition, data exchange and communication will be improved by implementation of comprehensive data models, which are able to supply all involved disciplines. To face these challenges, future CAD systems have to provide much more than the above mentioned “traditional” functionalities and data structures. As central data model as it has been during the years before, future-oriented CAD models have to include complete data structures, providing all product and process related information for effective cross-domain development of mechatronics systems. Besides geometrical, product structural, functional and production-related information of mechanics and hardware, this also comprises software-related requirements, structural and functional information.

Conclusions:

From the perspective of automotive industry, but not limited to this sector, future trends in CAD are influenced by demands for new approaches of efficient integration of geometry creation, functional development and verification, and that by simultaneous reduction of CAD program complexity. New technologies for holographic visualization and 3D printing of adaptable hardware models have potential to kick-off disruptive technology steps for product modeling, which might lead to completely new product styling and design processes. Implementation of knowledge into CAD models by problem-oriented specialization and generic programming will lead to more efficient development processes. A new challenge for future CAD represents the increasing product complexity caused by upcoming technologies, especially in the area of electrics and electronics. In any way, CAD models play a major role in the development of new cars and new technologies, today and in upcoming years.

References:

[1] Alesandra, B.; Cazzaniga, A.; Durelli, G.C.; Santambrogio, M.: An Open-Source Design and Validation Platform for Reconfigurable Systems, 17th International Conference on Field Programmable Logic and Applications, Norway, 2012.

[2] BRL-CAD Open Source Modelling, http://www.brlcad.org/, access date: 2015-12-30.

[3] Chaparala, R.T.; Hartman, N.W.; Springer, J.: Examining CAD interoperability through the use of ontologies, CAD’13, Bergamo, Italy, 2013.

[4] Claytronics Project Site, http://www.cs.cmu.edu/~claytronics/software/simulation.html, access date: 2015-12-29.

[5] Development Method for Mechatronic Systems, VDI Guideline 2206, The Association of German Engineers, Germany, 2003.

[6] Firebird, http://www.firebirdsql.org/, access date: 2015-04-13.

[7] Hirz, M.; Dietrich, W.; Gfrerrer, A.; Lang, J.: Integrated computer-aided design in automotive development: development processes, geometric fundamentals, methods of CAD, knowledge-based engineering data management, Springer, 2013. http://dx.doi.org/10.1007/978-3-642-11940-8.

[8] Jamieson P.; Kent K.B.; Gharibian F.; Shannon L.: Odin II - An Open-Source Verilog HDL Synthesis Tool for CAD Research, 18th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines, USA, 2010.

[9] Lohöfener, M.: Design of mechatronic systems and benefit of open source software tools, 9th International Workshop on Research and Education in Mechatronics, Italy, 2008.

[10] Linux, https://www.linux.com/, access date: 2015-12-30.

[11] MATLAB-MathWorks, http://de.mathworks.com/products/matlab/, access date: 2015-12-29.

[12] Microsoft Developer Network, VBScript, http://msdn.microsoft.com/de-at/dn308572.aspx, access date: 2015-12-29.

[13] Microsoft HoloLens, https://www.microsoft.com/microsoft-hololens, access date: 2015-12-29.

[14] On Any Device, https://www.onshape.com/, access date: 2015-12-29.

[15] Open CASCADE, http://www.opencascade.com/, access date: 2015-12-30.

[16] OpenOffice, https://www.openoffice.org/, access date: 2015-12-30.

[17] Salchner, M.; Stadler, S.; Hirz, M.; Mayr, J.; Ameye, J.: Multi-CAD Approach for Knowledge-Based, Computer-Aided Design and Applications, 2015, http://dx.doi.org/10.1080/16864360.2015.1131540.

[18] Sell, R.; Tamre, M.: Integration of V-model and SysML for advanced mechatronics system design, Int. Workshop on Research & Education in Mechatronics, At ANNECY, France, 2005.

[19] Terzidis, K.; Jungclaus, J.: Predicting the Future: Open Source CAAD?, Harvard University Publication, USA, 2004.