Quest Journals
Journal of Research in Mechanical Engineering
Volume 8 ~ Issue 4 (2022) pp: 01-08
ISSN(Online) : 2321-8185
www.questjournals.org

Research Paper

Development of Automotive Software

*Sujoy Kumar Saha
Mechanical Engineering Department
Indian Institute of Engineering Science and Technology Shibpur, Howrah, India
Corresponding Author: Sujoy Kumar Saha

ABSTRACT: *Requirements engineering is the description of functionality as well as the functional and non-functional textual documentation of property requirements of software being designed. This involves a phase of software development lifecycle and also a subdomain of software engineering. There are multiple forms; may be short textual descriptions of functionality as well as fully executable model-based (like V-model) specifications in designing automotive software systems. ISO 26262 prescribes such safety standard for the design in various phases in the automotive domain. Rapid changes in the development of modern automotive systems involve development methods; processes, and toolchains are constantly changed, modified, and improved to handle the increasing complexity of the development procedure. There are challenges in the development itself. The modification of the implied processes is to be dealt with carefully. Both textual and graphical overview of the main currently involved tools are important for optimal planning of development processes for future automotive systems. The number of features within a modern-day premium automobile has significantly increased and these are realized by software. The vehicle has to be kept on the track and may be displaying a movie for rear seat entertainment is a requirement. The software modules are on embedded systems, some of them fulfilling mission-critical task. Various tools and processes are to be integrated into the development process, delivering document metadata which can be used for insights like Software Fault Prediction (SFP). In this paper, requirements engineering for automotive embedded systems, status report on automotive software development and state=of-the-art tools and methods used in the automotive industry are discussed.*
KEYWORDS: *Double Pendulum, Numerical Solution, Simulation, System Behavior*

R*eceived 10 Apr, 2022;* ***R****evised 25 Apr, 2022;* ***A****ccepted 27Apr, 2022 © The author(s) 2022.*
P*ublished with open access at* ***www.questjournals.org***

I. INTRODUCTION

In this paper, the development of automotive software is discussed. Following topics are discussed: requirements engineering for automotive embedded systems, status report on automotive software development and state-of-the-art tools and methods used in the automotive industry are discussed. It is important to note the future trends of automotive software development.

II. REQUIREMENTS ENGINEERING FOR AUTOMOTIVE EMBEDDED SYSTEMS

In modern days, premium cars, trucks, buses and bikes may have more than 100 Electronic Control Units (ECUs) and software as much as 1 GB of onboard binary code in addition to maps, music and other downloadable data. There is enormous growth of software, which includes more safety-critical areas like collision avoidance, automatic parking and driving. Requirements engineering has methods, tools, and techniques for documenting, prioritizing, and maintaining quality of software. The user requirements, implicit expectations and professional standards are very important. 'Requirements' is an objective; a condition or capability to satisfy a contract, standard, specification, or other imposed documents. Testability of the system, traceability of the functionality to design and traceability of the project progress are the key issues. Software requirements engineering is becoming more and more important because the automotive industry has recognized the need to shift its innovation from the mechanical parts of the car to its electronics and software. These are just the two challenges which lead to two very important trends in automotive software requirements engineering. Car software designers or constructors gradually break down the requirements from the car level to the component level. They also gradually refine them from textual requirements to the behavior of the software. In the automotive domain, there are three tiers of suppliers.

III. STATUS REPORT OF THE AUTOMOTIVE SOFTWARE DEVELOPMENT

This section gives an overview of the current situation. There are challenges in the development process like currently used development tools and toolchains. The sophisticated development cycles and the expectations of the customers are hard to satisfy if no solutions for these challenges are found. Virtual methods for development and validation became more popular and feasible because of decreasing development cycles, test and validation at early stages of development, complexity of functions and systems, cost reduction, and additional issues. Many different approaches and tools try to address virtual development and validation methods. Interdivisional work often fails due to missing common toolchains and standardized processes. As solution for automotive applications, the Functional Engineering Platform (FEP) can be named. Conducting tests with *advanced driver-assistance systems (ADAS)* or automated driving vehicles at a certain point of the development process in the virtual environment is safer than in the real world. Highly connected vehicles require a large amount of data to perform automated driving. Additionally, driving data are used to improve vehicle functions and help to develop new systems. As hackers adjust and improve their approaches constantly, it may be necessary that the installed security mechanisms in the vehicle are updated frequently. In the automotive domain, several tools and toolchains have been established over the decades. In the current automotive development projects, the MATLAB/Simulink/TargetLink toolchain is widely used for system design, implementation and code generation. IBM Rational Rhapsody is a UML modeling tool supports all types of UML diagrams with the corresponding elements and provides simulation features and code generation techniques.

IV. TOOLS AND METHODS USED IN AUTOMTIVE INDUSTRY

A modern-day premium car can contain more than a 100 Electronic Control Units (ECUs), 10 networks and up to 2 SIM cards, and several kilometers of cable. The software within a modern-day premium car might claim up to one million Lines of Code (LOC). Software modules need to fulfill various tasks. Some features require multiple sensor values connected to different ECUs and influence more than one actuator. The majority of software components need to run under embedded conditions and deliver their result within real time and soft real time. Some features need to meet guidelines like that of Motor Industry Software Reliability Association (MISRA) or need to satisfy safety requirements as Automotive Safety Integrity Level (ASIL). Therefore, a car can be seen as a heterogeneous network of many distributed computers running a high number of software functions, either stand-alone, distributed asynchronous, or synchronous. A core component is a clear separation between requirements, development, and testing realized by using different specialized tools working together with an adopted toolchain. This toolchain enables exporting requirements written in DOORS into a Matlab Simulink model. Thus the engineer designing the model can trace if subsystems accord to requirements. The developed taxonomy guides engineers on the selection of new tools and methods. Software development has various sub-milestones like interface freeze and feature freeze. Software Fault Prediction (SFP) is a new idea to be integrated. Automotive software development follows restrictive settings in terms of coding guidelines. It is possible to gain advantages. Support Vector Machine (SVM), Naïve Bayes (NB), and Random Forest (RF) repeat the measurement step and use the trained machine learning system to derive a probability if there is a bug.

V. CONCLUSION

In this paper, requirements engineering for automotive embedded systems, status report on automotive software development and state-of-the-art tools and methods used in the automotive industry have been discussed. The readers' attention is drawn to the references for more information.

REFERENCES

[1]. Leveson, N.G., Safeware, System Safety and Computers. 1995, Reading, Mass.: Addison Wesley Company.
[2]. Simon, H.A., Science of the Artificial. 1981, MIT Press, Cambridge.
[3]. Suri, N., Walter, C.J. and Hugue, M.M., eds., Advances in Ultra-Dependable Systems. 1995, IEEE Press.
[4]. Kopetz, H. and Bauer, G., The Time-Triggered Architecture. Proceedings of the IEEE, 2003. 91 (January 2003).
[5]. J. Schaeuffele and T. Zurawka, Automotive Software Engineering, 2nd ed., SAE Int'l, 2016.
[6]. P. Koopman and M. Wagner, "Autonomous Vehicle Safety: An Interdisciplinary Challenge," IEEE Intelligent Transportation Systems Magazine, vol. 9, no. 1, 2017, pp. 90–96.
[7]. C. Ebert, "Implementing Functional Safety," IEEE Software, vol. 32, no. 5, 2015, pp. 84–89.
[8]. Bosch Automotive Electrics and Automotive Electronics: Systems and Components, Networking and Hybrid Drive, 5th ed., Springer, 2014.
[9]. Jesper Andersson: Die UML echtzeitfähig Machen mit der formalen Sprache SDL, OBJEKTspektrum 3, 1999.
[10]. Morgan Björkander: Graphical Programming Using UML and SDL, IEEE Computer, Vol. 24, 2001.
[11]. Peter Braun and Martin Rappl: Model based Systems Engineering - A Unified Approach using UML, Systems Engineering - A Key to Competitive Advantage for All Industries Proceedings of the 2nd European Systems Engineering Conference (EuSEC 2000), Herbert Utz Verlag GmbH, München, 2000.
[12]. Peter Braun, Martin Rappl and Jörg Schäuffele: Software entwicklungen Phase, VDI-Berichte, Nr. 1547, S. 265 ff., 2000.

[13]. Brodsky, Clark, Cook, Evans and Kent: Feasability Study in Rearchitecting the UML as a Family of Languages Using a Precise OO Meta-Modeling Approach, The pUML Group, 2000.
[14]. A.-P. Bröhl and W. Dröschel: Das V-Modell – Der Standard für die Softwareentwicklung mit Praxisleitfaden, R. Oldenburg Verlag München Wien 1993
[15]. Manfred Broy ++: The Design of distributed Systems, An introduction to FOCUS – Revised Version, Technical Report, TUM-I9202, Technische Universität München, 1993.
[16]. Manfred Broy, Michael von der Beeck, Peter Braun and Martin Rappl: A fundamental critique of the UML for the specification of embedded systems.
[17]. Pio Torre Flores ++: Integration of a Structuring Concept for Vehicle Control Systems into the Software Development Process using UML Modeling Methods, SAE Technical Paper Series 2001-01-0066, Detroit, 2001.
[18]. Bernd Gebhard and Martin Rappl: Requirements Management for Automotive Systems Development, SAE Technical Paper Series 2000-01-0716, Detroit, 2000.
[19]. C. Hansen, O. Bringmann and W. Rosenstiel: A VHDL Component Model for Mixed Abstraction Level Simulation and Behavioral Synthesis, Proceedings of Forum on Design Languages (FDL'99), Lyon, Frankreich, September 1999.
[20]. Derek Hatley and Imtiaz Pirbhai: Strategies for real time system specification, Dorset House Publishers, New York, 1988.
[21]. Object Management Group: OMG Unified Modeling Language Specification (draft), Version 1.3 alpha R5, March 1999.
[22]. OMG: XML Metadata Interchange (XMI), Proposal to the OMG OA&DTF RFP 3: Stream-based Model Interchange Format (SMIF), OMG Dokument, Oct 20th, 1998.
[23]. Bernhard Rumpe and Andy Schürr: UML + ROOM as a Standard ADL?, Engineering of Complex Computer Systems, ICECCS'99 Proceedings, IEEE Computer Society, 2000.
[24]. Bran Selic, Garth Gullekson and Paul T. Ward: Real Time Object Oriented Modeling, John Wiley, 1994.
[25]. Desmond F. D'Souza and Alan C. Wills: Objects, Components and Frameworks with UML – the CATALYSIS approach, Addison-Wesley, 1998.
[26]. Oscar Slotosch ++: Tool supported Specification and Simulation of Distributed Systems, Proceedings International Symposium on Software Engineering for Parallel and Distributed Systems, pp. 155-164, IEEE Computer Society, Los Alamitos, California, 1998.
[27]. Richard Stevens ++: systems engineering – coping with complexity, Prentice Hall Europe, Hertfordshire, 1998.
[28]. Andrew S. Tanenbaum: Computer-Netzwerke, Wolfram's Verlag, 2. Auflage, 1992.
Dubois H, Peraldi-Frati MA, Lakhal F (2010) A model for requirements traceability in a heterogeneous model-based design process: application to automotive embedded systems. In: 2010 15th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS). IEEE, Piscataway, pp 233–242
[29]. Eliasson U, Heldal R, Lantz J, Berger C (2014) Agile model-driven engineering in mechatronic systems-an industrial case study. In: Model-driven engineering languages and systems. Springer, Cham, pp 433–449
[30]. Eliasson U, Heldal R, Knauss E, Pelliccione P (2015) The need of complementing plan-driven requirements engineering with emerging communication: experiences from volvo car group. In: 2015 IEEE international Requirements Engineering conference (RE). IEEE, Piscataway, pp 372–381
[31]. Hardt M, Mackenthun R, Bielefeld J (2002) Integrating ECUs in vehicles-requirements engineering in series development. In: 2002 IEEE international Requirements Engineering conference (RE). IEEE, Piscataway, pp 227–236
[32]. Honig WL (2016) Requirements metrics - definitions of a working list of possible metrics for requirements quality. Retrieved from Loyola eCommons, Computer Science: Faculty Publications and Other Works
[33]. Houdek F (2013) Managing large scale specification projects. In: 19th international working conference on Requirements Engineering Foundations for Software Quality, REFSQ 2013, Essen, Germany, 8–11 April 2013
[34]. IEEE (1990) IEEE standard glossary of software engineering terminology (IEEE std 610.12-1990). IEEE Computer Society, Los Alamitos
[35]. ISO I (2011) 26262–road vehicles-functional safety. International Standard ISO/FDIS 26262
[36]. Jacobson I, Booch G, Rumbaugh J (1997) The objectory software development process. Addison Wesley, Boston. ISBN: 0-201-57169-2
[37]. Langenfeld V, Post A, Podelski A (2016) Requirements defects over a project lifetime: an empirical analysis of defect data from a 5-year automotive project at Bosch. In: Requirements engineering: foundation for software quality. Springer, Cham, pp 145–160
[38]. Mahally MM, Staron M, Bosch J (2015) Barriers and enablers for shortening software development lead-time in mechatronics organizations: a case study. In: Proceedings of the 2015 10th joint meeting on foundations of software engineering. ACM, New York, pp 1006–1009
[39]. Mahmud N, Seceleanu C, Ljungkrantz O (2015) Resa: an ontology-based requirement specification language tailored to automotive systems. In: 10th IEEE international Symposium on Industrial Embedded Systems (SIES), 2015. IEEE, Piscataway, pp 1–10
[40]. Manhart P, Schneider K (2004) Breaking the ice for agile development of embedded software: an industry experience report. In: Proceedings of the 26th international conference on software engineering. IEEE Computer Society, Washington, pp 378–386
[41]. Mellegård N, Staron M (2008) Methodology for requirements engineering in model-based projects for reactive automotive software. In: 18th ECOOP doctoral symposium and PhD student workshop, p 23
[42]. Mellegård N, Staron M (2009) A domain specific modelling language for specifying and visualizing requirements. In: The first international workshop on domain engineering, DE@ CAiSE, Amsterdam
[43]. Mellegård N, Staron M (2010) Characterizing model usage in embedded software engineering: a case study. In: Proceedings of the fourth European conference on software architecture: companion volume. ACM, New York, pp 245–252
[44]. Mellegård N, Staron M (2010) Distribution of effort among software development artefacts: an initial case study. In: Enterprise, business-process and information systems modeling. Springer, Berlin, pp 234–246
[45]. Mellegård N, Staron M (2010) Improving efficiency of change impact assessment using graphical requirement specifications: an experiment. In: Product-focused software process improvement. Springer, Berlin, pp 336–350
[46]. Ott D (2012) Defects in natural language requirement specifications at mercedes-benz: an investigation using a combination of legacy data and expert opinion. In: 2012 20th IEEE international Requirements Engineering conference (RE). IEEE, Piscataway, pp 291–296
[47]. Ott D (2013) Automatic requirement categorization of large natural language specifications at mercedes-benz for review improvements. In: Requirements engineering: foundation for software quality. Springer, Berlin, pp 50–64
[48]. Peraldi-Frati MA, Albinet A (2010) Requirement traceability in safety critical systems. In: Proceedings of the 1st workshop on critical automotive applications: robustness & safety. ACM, New York, pp 11–14
[49]. Pernstål J, Gorschek T, Feldt R, Florén D (2013) Software process improvement in interdepartmental development of software-intensive automotive systems–a case study. In: Productfocused software process improvement. Springer, Berlin, pp 93–107

[50]. Rana R, Staron M, Berger C, Hansson J, Nilsson M, Törner F (2013) Improving fault injection in automotive model based development using fault bypass modeling. In: GI-Jahrestagung. Chalmers University of Technology, Gothenburg, pp 2577–2591
[51]. Rana R, Staron M, Mellegård N, Berger C, Hansson J, Nilsson M, Törner F (2013) Evaluation of standard reliability growth models in the context of automotive software systems. In: Product-focused software process improvement. Springer, Berlin, pp 324–329
[52]. Rana R, Staron M, Berger C, Hansson J, Nilsson M, Törner F (2013) Increasing efficiency of ISO 26262 verification and validation by combining fault injection and mutation testing with model based development. In: ICSOFT 2013, pp 251–257
[53]. Sagstetter F, Lukasiewycz M, Steinhorst S, Wolf M, Bouard A, Harris WR, Jha S, Peyrin T, Poschmann A, Chakraborty S (2013) Security challenges in automotive hardware/software architecture design. In: Proceedings of the conference on design, automation and test in Europe, EDA consortium, pp 458–463
[54]. Siegl S, Russer M, Hielscher KS (2015) Partitioning the requirements of embedded systems by input/output dependency analysis for compositional creation of parallel test models. In: 9th annual IEEE international Systems Conference (SysCon), 2015. IEEE, Piscataway, pp 96–102
[55]. SimulinkDemo (2012) Modeling an anti-lock braking system. The MathWorks, Inc, Natick. Copyright 2005-2010
[56]. Sinha P (2011) Architectural design and reliability analysis of a fail-operational brake-by-wire system from iso 26262 perspectives. Reliab Eng Syst Saf 96(10), 1349–1359
[57]. Staron M (2017) Automotive software architectures: an introduction. Springer, Cham
[58]. Törner F, Ivarsson M, Pettersson F, Öhman P (2006) Defects in automotive use cases. In: Proceedings of the 2006 ACM/IEEE international symposium on empirical software engineering. ACM, New York, pp 115–123
[59]. Vogelsanag A, Fuhrmann S (2013) Why feature dependencies challenge the requirements engineering of automotive systems: an empirical study. In: 2013 21st IEEE international Requirements Engineering conference (RE). IEEE, Piscataway, pp 267–272
[60]. Weber M, Weisbrod J (2002) Requirements engineering in automotive developmentexperiences and challenges. In: 2002 IEEE international Requirements Engineering conference (RE). IEEE, Piscataway, pp 331–340
[61]. Wright A (2011) Hacking cars. Commun ACM 54(11):18–19
[62]. Ali NM, Hosking J, Grundy J (2013) A taxonomy and mapping of computer-based critiquing tools. IEEE Trans Softw Eng 39(11):1494–1520
[63]. Azuma M, Coallier F, Garbajosa J (2003) How to apply the bloom taxonomy to software engineering. In: Eleventh annual international workshop on software technology and engineering practice, 2003. IEEE, Piscataway
[64]. Babar MA, Gorton I (2004) Comparison of scenario-based software architecture evaluation methods. In: 11th Asia-Pacific software engineering conference, 2004. IEEE, Piscataway
[65]. Blom H, Hagl F, Papadopoulos Y, Reiser MO, Sjöstedt CJ, Chen DJ, Kolagari R (2012) EAST-ADL - an architecture description language for automotive software-intensive systems. International Standard
[66]. Blum BI (1994) A taxonomy of software development methods. Commun ACM 37(11):82–94
[67]. Bock F, Homm D, Siegl S, German R (2016) A taxonomy for tools, processes and languages in automotive software engineering. In: Zizka J, Nagamalai D (eds) Computer science & information technology. AIRCC Publishing Corporation, Chennai
[68]. Bock F, Sippl C, German R (2017) Fully automated vehicles: challenges, expectations and methods. In: Bargende M, Reuss HC, Wiedemann J (eds) Proceedings of 17th Internationales Stuttgarter Symposium: Automobil- und Motorentechnik, Stuttgart, Germany
[69]. Bröhl A (1993) The V-model. In: Software - application delevopment - information systems. Oldenbourg, Munich
[70]. Broy M, Feilkas M, Herrmannsdoerfer M, Merenda S, Ratiu D (2010) Seamless model-based development: from isolated tools to integrated model engineering environments. Proc IEEE 98(4):526–545
[71]. Caspi P, Pilaud D, Halbwachs N, Plaice JA (1987) LUSTRE: a declarative language for realtime programming. In: Proceedings of the 14th ACM SIGACT-SIGPLAN symposium on principles of programming languages, New York, NY, USA, POPL '87
[72]. Cheu RL, Tan Y, Lee D (2003) Comparison of paramics and GETRAM/AIMSUN microscopic traffic simulation tools. In: 83rd annual meeting of the transportation research board
[73]. Dajsuren Y, van den Brand MG, Serebrenik A, Roubtsov S (2013) Simulink models are also software: modularity assessment. In: Proceedings of the 9th international ACM sigsoft conference on quality of software architectures. ACM, New York, pp 99–106
[74]. Di Natale M, Sangiovanni-Vincentelli AL (2010) Moving from federated to integrated architectures in automotive: the role of standards, methods and tools. Proc IEEE 98(4):603–620
[75]. Djanatliev A, Dulz W, German R, Schneider V (2011) Veritas - a versatile modelling environment for test-driven agile simulation. In: Proceedings of the 2011 winter simulation conference, Phoenix, AZ, USA, WSC 2011
[76]. Eckhoff D (2016) Simulation of privacy-enhancing technologies in vehicular ad-hoc networks. PhD thesis, University of Erlangen
[77]. Fürst S, Bechter M (2016) Autosar for connected and autonomous vehicles: the autosar adaptive platform. In: 2016 46th annual IEEE/IFIP international conference on Dependable Systems and Networks Workshop (DSN-W)
[78]. Fürst S, Mössinger J, Bunzel S, Weber T, Kirschke-Biller F, Heitkämper P, Kinkelin G, Nishikawa K, Lange K (2009) Autosar–a worldwide standard is on the road. In: 14th international VDI congress electronic systems for vehicles, Baden-Baden, vol 62
[79]. Guo Y, Jones RP (2009) A study of approaches for model based development of an automotive driver information system. In: 3rd annual IEEE systems conference, 2009. IEEE, Piscataway
[80]. Hoffmann H (2014) Systems engineering best practices with the rational solution for systems and software engineering
[81]. IO for Standardization (2009) ISO/DIS 26262-1 - Road vehicles - functional safety - part 1 glossary
[82]. Kielar PM, Biedermann DH, Borrmann A (2016) MomenTUMv2: a modular, extensible, and generic agent-based pedestrian behavior simulation framework. TUM-I1643, Technische Universität München
[83]. Klauda M, Hamann R, Kriso S (2013) ISO 26262 – Muss das Rad neu erfunden werden?, Springer Fachmedien Wiesbaden, Wiesbaden, pp 224–227
[84]. Knauss A, Schröder J, Berger C, Eriksson H (2017) Paving the roadway for safety of automated vehicles: an empirical study on testing challenges. In: 2017 IEEE Intelligent Vehicles symposium (IV)
[85]. Kornecki AJ, Zalewski J (2003) Design tool assessment for safety-critical software development. In: Conference: software engineering workshop, 2003
[86]. Kotusevski G, Hawick K (2009) A review of traffic simulation software. Res Lett Inf Math Sci. 13
[87]. Lachmann R, Schaefer I (2013) Herausforderungen beim Testen von Fahrerassistenzsystemen. In: GI-Jahrestagung
[88]. Mubasher MM, ul Qounain JSW (2015) Systematic literature review of vehicular traffic flow simulators. In: 2015 international conference on Open Source Software Computing (OSSCOM)
[89]. Rasshofer RH, Gresser K (2005) Automotive radar and lidar systems for next generation driver assistance functions. Adv Radio Sci 3:205–209

[90]. Ronaldo A, Ismail T (2012) Comparison of the two micro-simulation software AIMSUN & SUMO for highway traffic modelling, Linköping University, Communications and Transport Systems, The Institute of Technology, p 96. http://www.diva-portal.org/smash/get/diva2: 555913/FULLTEXT01.pdf
[91]. Schneider V, German R (2013) Integration of test-driven agile simulation approach in serviceoriented tool environment. In: Proceedings of the 46th annual simulation symposium, San Diego, CA, USA, ANSS 2013
[92]. Alves V, Matos P, Cole L, Vasconcelos A, Borba P, Ramalho G (2007) Extracting and evolving code in product lines with aspect-oriented programming. In: Transactions on aspect-oriented software development IV. Springer, Berlin, pp 117–142
[93]. Apel S, Kästner C (2009) An overview of feature-oriented software development. J Object Technol 8(5):49–84
[94]. Bak K, Czarnecki K, W ˛asowski A (2011) Feature and meta-models in Clafer: mixed, specialized, and coupled. In: Proceedings of the international conference on software language engineering (SLE), SLE '11. Springer, Berlin, pp 102–122
[95]. Batory D (2004) Feature-oriented programming and the AHEAD tool suite. In: Proceedings of the international conference on software engineering (ICSE), ICSE '04. IEEE, Piscataway, pp 702–703
[96]. Berger T, Rublack R, Nair D, Atlee JM, Becker M, Czarnecki K, W ˛asowski A (2013) A survey of variability modeling in industrial practice. In: Proceedings of the international workshop on variability modeling in software-intensive systems (VaMoS), VaMoS '13. ACM, New York, pp 7:1–7:8
[97]. Berger T, Lettner D, Rubin J, Grünbacher P, Silva A, Becker M, Chechik M, Czarnecki K (2015) What is a feature? a qualitative study of features in industrial software product lines. In: Proceedings of the international software product line conference (SPLC), SPLC '15. ACM, New York, pp 16–25
[98]. Beuche D (2012) Modeling and building software product lines with pure:variants. In: Proceedings of the international software product line conference (SPLC), SPLC '12. ACM, New York, pp 255–255
[99]. Clements PC, Northrop LM (2001) Software product lines: practices and patterns. AddisonWesley, Boston
[100]. Czarnecki K, Eisenecker UW (2000) Generative programming: methods, tools, and applications. Addison-Wesley, Boston
[101]. Damiani F, Lienhardt M (2016) On type checking delta-oriented product lines. In: Proceedings of the international conference on integrated formal methods (iFM), iFM '16. Springer, Berlin, pp 47–62
[102]. Damiani F, Lienhardt M (2016) Refactoring delta oriented product lines to enforce guidelines for efficient type-checking. In: Proceedings of the international symposium on leveraging applications of formal methods, verification and validation (ISoLA), ISoLA'16. Springer, Berlin
[103]. Dubinsky Y, Rubin J, Berger T, Duszynski S, Becker M, Czarnecki K (2013) An exploratory study of cloning in industrial software product lines. In: Proceedings of the European conference on software maintenance and reengineering (CSMR), CSMR '13. IEEE, Piscataway, pp 25–34
[104]. Figueiredo E, Cacho N, Sant'Anna C, Monteiro M, Kulesza U, Garcia A, Soares S, Ferrari F, Khan S, Dantas F (2008) Evolving software product lines with aspects. In: Proceedings of the international conference on software engineering (ICSE), ICSE '08. IEEE, Piscataway, pp 261–270
[105]. Greenfield J, Short K (2003) Software factories: assembling applications with patterns, models, frameworks and tools. In: Proceedings of the international conference on object-oriented programming, systems, languages and applications (OOPSLA), OOPSLA '03. ACM, New York, pp 16–27
[106]. Hackers remotely kill a jeep on the highway – with me in it. WIRED. http://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/
[107]. Remote exploitation of an unaltered passenger vehicle. Dr. Charlie Miller and Chris Valasek. http://illmatics.com/Remote%20Car%20Hacking.pdf
[108]. Sens. Markey, blumenthal introduce legislation to protect drivers from auto security, privacy risks with standards & "cyber dashboard" rating system.http://www.markey.senate.gov/news/press-releases/sens-markey-blumenthal-introduce-legislation-to-protect-drivers-from auto security-privacy-risks-with-standards-and-cyber-dashboard-rating-system
[109]. Tracking & hacking: security & privacy gaps put American drivers at risk. Ed Markey, United States Senator for Massachusetts; February2015.http://www.markey.senate.gov/imo/media/doc/2015-02-06_MarkeyReportTracking_Hacking_CarSecurity%202.
[110]. The STRIDE threat model. https://msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx
[111]. Hacking a Tesla model S: what we found and what we learned. Lookout Blog. https://blog.lookout.com/blog/2015/08/07/hacking-a-tesla/
[112]. The 'Heartbleed' security flaw that affects most of the Internet. CNN.com. http://edition.cnn.com/2014/04/08/tech/web/heartbleed-openssl/
[113]. How the poodle computer bug impacts business? Fortune.com. http://fortune.com/2014/11/12/poodle-bug/
[114]. FREAKing hell: all windows versions vulnerable to SSL snoop. The Register. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-0204
[115]. Vehicle electrical system security committee. http://www.sae.org/works/committeeHome.do?comtID=TEVEES18
[116]. Birolini, Alessandro, Reliability Engineering. Theory and Practice, Springer-Verlag, Berlin, Heidelberg, New York, 1999.
[117]. Birolini, Alessandro, Zuverlässigkeit von Geräten und Systemen, Springer-Verlag, Berlin, Heidelberg, New York, 1997.
[118]. Ehrenberger, Wolfgang, Software-Verifi kation: Verfahren für den Zuverlässigkeitsnachweis von Software, Carl Hanser Verlag, Munich, Vienna, 2002.
[119]. U.S. Environmental Protection Agency, Control of Air Pollution from Motor Vehicles and New Motor Vehicles; Modifi cation of Federal On-Board Diagnostic Regulations for Light-Duty Vehicles and Light-Duty Trucks; Extension of Acceptance of California OBD II Requirements, December 1998.
[120]. Halang, W.A., and Konakovsky, R., Sicherheitsgerichtete Echtzeitsysteme, R. Oldenbourg Verlag, Munich, Vienna, 1999.
[121]. Lin, Shu, and Costello, Daniel J., Error Control Coding, Prentice-Hall, Englewood Cliffs, NJ, 1983.
[122]. Leveson, Nancy G., Safeware. System Safety and Computers. A Guide to Preventing Accidents and Losses Caused by Technology, Addison-Wesley, New York, 1995.
[123]. Storey, Neil, Safety-Critical Computer Systems, Prentice-Hall, Harlow, England, 1996.
[124]. Deutsches Institut für Normung e.V., DIN 25448—Ausfalleffektanalyse (Fehler-Möglichkeits- und Einfl uss-Analyse), May 1990.
[125]. "Der neue BMW 7er," Automobiltechnische Zeitschrift/Motortechnische Zeitschrift, ATZ/MTZ Extra (special issue), November 2001.
[126]. International Organization for Standardization, ISO 11519: Straßenfahrzeuge—Serielle Datenübertragung mit niedriger Übertragungsrate, 1994.
[127]. Bytefl ight, www.bytefl ight.de.
[128]. MOST Media Orientated System Transport, www.mostcooperation.com.

[129]. "Die neue Mercedes-Benz-E-Klasse," Automobiltechnische Zeitschrift/Motortechnische Zeitschrift, ATZ/MTZ Extra (special issue), May 2002.
[130]. Bluetooth, www.bluetooth.com.
[131]. LIN Local Interconnect Network, www.lin-subbus.de.
[132]. Boy, J., Dudek, C., and Kuschel, S., Projektmanagement. Grundlagen, Methoden und Techniken, Zusammenhänge, Gabal-Verlag, Offenbach, Germany, 1998.
[133]. The Motor Industry Software Reliability Association, Development Guidelines for Vehicle Based Software, 1994, http://www.misra.org.uk.
[134]. International Council on Systems Engineering, http://www.incose.org.
[135]. Balzert, Helmut, Lehrbuch der Software-Technik, 2nd Edition, Spektrum Akademischer Verlag, Heidelberg—Berlin, Germany, 2000.
[136]. ETAS GmbH (Ed.), ASCET V5.0 User's Guide. ETAS GmbH, Stuttgart, Germany, 2004.
[137]. Selic, B., Gullekson, G., and Ward, P.T., Real-Time Object-Oriented Modeling, John Wiley & Sons, New York, 1994.
[138]. Stoer, Josef, Numerische Mathematik 1, 8th Edition, Springer-Verlag, Berlin, Heidelberg, New York, 1999.
[139]. Kernighan, B.W., and Ritchie, D.M., Programmieren in C, 2nd Edition, Carl Hanser Verlag, Munich, Germany, 1990.
[140]. Broy, Manfred, Informatik. Eine grundlegende Einführung, Vols. 1/2, Springer-Verlag, Berlin, Heidelberg, New York, 1998.
[141]. Wirth, N., Grundlagen und Techniken des Compilerbaus, Addison-Wesley, Bonn, Germany, and Paris, France, 1996.
[142]. Manufacturer Supplier Relationship, Working Groups MEGMA and MEDOC, http://www.msr-wg.de.
[143]. The Motor Industry Software Reliability Association, Guidelines for the Use of the C Language in Vehicle-Based Software, 1998.
[144]. van Basshuysen, Richard, and Schäfer, Fred (Eds.), Handbuch Verbrennungsmotor. Grundlagen, Komponenten, Systeme, Perspektiven, 1st Edition, Vieweg-Verlag,
[145]. Wiesbaden, Germany, 2002; also published as van Basshuysen, Richard, and Schäfer, Fred (Eds.), Internal Combustion Engine Reference Handbook, SAE International, Warrendale, PA, 2004.
[146]. Pauli, B., and Meyna, A., "Zuverlässigkeitsprognosen für elektronische Steuergeräte im Kraftfahrzeug" ("Reliability of Electronic Control Units in Motor Vehicles"; orginal German), in Proceedings of the 7th International Congress "Electronic Systems for
[147]. Vehicles," Baden-Baden, Germany, September 13, 1996, ed. by VDI Society Vehicle and Transportation Technology, VDI-Berichte 1287, VDI-Verlag, Düsseldorf, Germany, 1996, pp. 87–105.
[148]. Beer, A., and Schmidt, M., "Funktionale Sicherheit Sicherheitsrelevanter Systeme im Kraftfahrzeug" ("Functional Safety of Safety Relevant Systems in Vehicles"; original German), in Proceedings of the 9th International Congress "Electronic Systems for Vehicles," Baden-Baden, Germany, October 5–6, 2000, ed. by VDI Society Vehicle and Transportation Technology, VDI-Berichte 1547, VDI-Verlag, Düsseldorf, Germany, 2000, pp. 391–409.
[149]. UML Unifi ed Modeling Language™, www.uml.org.
[150]. Harel, D., Statecharts. A Visual Formalism for Complex Systems, Science of Computer Programming, Vol. 8, Elsevier Science Publishers, North Holland, 1987.
[151]. ETAS GmbH (Ed.), ERCOSEK V4.2 User's Guide, ETAS GmbH, Stuttgart, Germany, 2002.
[152]. ETAS GmbH (Ed.), INCA V5.0 User's Guide, ETAS GmbH, Stuttgart, Germany, 2004.
[153]. Grams, T., Denkfallen und Programmierfehler, Springer-Verlag, Berlin, Heidelberg, New York, 1990.
[154]. ETAS GmbH (Ed.), Data Declaration System V2.3 User's Guide, ETAS GmbH, Stuttgart, Germany, 2001.
[155]. ETAS GmbH (Ed.), LABCAR-OPERATOR V2.0 User's Guide, ETAS GmbH, Stuttgart, Germany, 2003.
[156]. Kühner, T., Seefried, V., Litschel, M., and Schelling, H., "Realisierung Virtueller Fahrzeugfunktionen für Vernetzte Systeme auf Basis Standardisierter Software-Bausteine" ("Implementation of Virtual Vehicle Functions for Networked Systems Using Standardized Software Modules"; original German), in Proceedings of the 7th International Congress "Electronic Systems for Vehicles," Baden-Baden, Germany, September 12–13, 1996, ed. by VDI Society Vehicle and Transportation Technology, VDI-Berichte 1287, VDI-Verlag, Düsseldorf, Germany, 1996, pp. 691–708.
[157]. Institute of Electrical and Electronics Engineers: NEXUS, www.ieee-isto.org/ Nexus5001.
[158]. Institute of Electrical and Electronics Engineers: JTAG IEEE 1149.1, www.ieee.org.
[159]. ETAS GmbH (Ed.), ETK S2.0 Emulator Probe for Serial Debug Interfaces, Data Sheet, ETAS GmbH, Stuttgart, Germany, 2002.
[160]. ETAS GmbH (Ed.), ETK 7.1 16-Bit Emulator Probe, Data Sheet, ETAS GmbH, Stuttgart, Germany, 2001.
[161]. Gumpinger, F., Huber, F.-M., and Siefermann, O., "BMW Car & Key Memory: Der Kunde bekommt sein individuelles Fahrzeug" ("BMW Car & Key Memory—The Customer Will Receive His Individual Vehicle"; original German), in Proceedings of the 8th International Congress "Electronic Systems for Vehicles," Baden-Baden, Germany, October 8–9, 1998, ed. by VDI Society Vehicle and Transportation Technology, VDI-Berichte 1415, VDIVerlag, Düsseldorf, Germany, 1998, pp. 995–1007.
[162]. Singh, S., Geheime Botschaften. Die Kunst der Verschlüsselung von der Antike bis in die Zeiten des Internet, Deutscher Taschenbuch Verlag, Munich, Germany, 2001; also published as Singh, S., The Code Book, Fourth Estate, ISBN 1857028899, 2000.
[163]. Birla, S.: Challenge problems for model-based integration of embedded systems (MoBIES. DTIC AFRL-IF-WP-TR-2004-1523 (2004)
[164]. Torngren, M., Larses, O.: Characterization of model based development of embedded control systems from a mechatronic perspective: drivers, processes, technology and their maturity. Technical Report TRITA-MMK 2004:23, Mechatronics Lab, Department of Machine Design, Royal Institute of Technoglogy, KTH, Stockholm, Sweden (2004)
[165]. Mellor, S.J., Clark, A.N., Futagami, T.: Model-driven development. IEEE Software 20(5), 14–18 (2003)
[166]. Price, D.: Ap233 state machine support (2005), http://www.ap233.org/ModuleSets/Behavior/AP233 State Machine 2005-09-15 update.zip/view
[167]. Object Management Group: Uml for systems engineering, request for proposal (2003), http://www.sysml.org/artifacts/refs/UML-for-SE-RFP.pdf
[168]. Object Management Group: Unified modeling language: Superstructure, version 2.0 (2005), http://www.omg.org/docs/formal/05-07-04.pdf
[169]. International Society for Automotive Engineers (SAE) AADL Team: Architecture analysis & design language (AADL) standard (2004), http://www.aadl.info
[170]. EAST-EEA Partners: East-eea architecture description language, version 1.0.2 (2004), http://www.east-eea.net/start.asp
[171]. Hamon, G., Rushby, J.: An Operational Semantics for Stateflow. In: Wermelinger, M., Margaria-Steffen, T. (eds.) FASE 2004. LNCS, vol. 2984, pp. 229–243. Springer, Heidelberg (2004)
[172]. Anton, J., da Costa, P., Errington, L.: Formal synthesis of generators for embedded systems. Technical report, Kestrel Technology, Palo Alto, CA (2005)

[173]. Arango, G.: Domain Analysis. In: Marciniak, J. (ed.) Encyclopedia of Software Engineering, vol. 1, pp. 424–434. Wiley, Chichester (1994)
[174]. The Software Engineering Institute (SEI): A framework for software product line practice version 4.2 (2005), http://www.sei.cmu.edu/productlines/framework.html
[175]. Chen, K., Sztipanovits, J., Neema, S.: A case study on semantic unit composition. In: MISE 2007: Proceedings of the International Workshop on Modeling in Software Engineering, Washington, DC, USA, p. 3. IEEE Computer Society Press, Los Alamitos (2007)
[176]. Villa, T., Kam, T., Brayton, R.K., Sangiovanni-Vincentelli, A.L.: Synthesis of Finite State Machines: logic Optimization. Kluwer Academic Publishers, Dordrecht (1997)
[177]. Berry, G., Gonthier, G.: The synchronous programming language esterel: design,semantics, implementation. Technical Report 842, INRIA (1988)
[178]. Halbwachs, N., Caspi, P., Raymond, P., Pilaud, D.: The synchronous dataflow programming language Lustre. Proceedings of the IEEE 79, 1305–1320 (1991)15. Deshpande, A., G¨ollu, A., Varaiya, P.: SHIFT: a formalism and a programming language for dynamic networks of hybrid automata. In: HS 1997. LNCS, vol. 1567, Springer, Heidelberg (1996)
[179]. Henzinger, T., Kirsch, C., Sanvido, M., Pree, W.: From control models to real-time code using Giotto. IEEE Control Systems Magazine (2003)
[180]. Karsai, G., Sztipanovits, J., Ledeczi, A., Bapty, T.: Model-integrated development of embedded software. In: Proceedings of the IEEE, vol. 91, pp. 145–164 (2003)
[181]. Model-Driven Hybrid and Embedded Software for Automotive Applications. In: 2nd RTAS Workshop on Model-Driven Embedded Systems (MoDES 2004) (2004)
[182]. Shah, B., Dennison, R., Gray, J.: A model-driven approach for generating embeddedrobot navigation control software. In: ACM-SE 42: Proceedings of the 42nd annual Southeast regional conference, pp. 332–335. ACM Press, New York (2004)
[183]. Stauner, T.: Discrete-Time Refinement of Hybrid Automata. In: Tomlin, C.J., Green street, M.R. (eds.) HSCC 2002. LNCS, vol. 2289, pp. 407–420. Springer, Heidelberg (2002)
[184]. Hybrid Toolbox - Hybrid Systems, Control, Optimization, http://www.dii.unisi.it/hybrid/toolbox
[185]. Henzinger, T.A., Ho, P.H., Wong-Toi, H.: HYTECH: A model checker for hybridsystems. International Journal on Software Tools for Technology Transfer 1, 110–122 (1997)
[186]. Asarin, E., Dang, T., Maler, O.: The d/dt Tool for Verification of Hybrid Systems. In: Brinksma, E., Larsen, K.G. (eds.) CAV 2002. LNCS, vol. 2404, pp. 365–370. Springer, Heidelberg (2002)
[187]. Tools, H.S.: http://wiki.grasp.upenn.edu/graspdoc/hst/
[188]. Hur, Y., Kim, J., Lee, I., Choi, J.Y.: Sound Code Generation from Communicating Hybrid Models. In: Alur, R., Pappas, G.J. (eds.) HSCC 2004. LNCS, vol. 2993, pp. 432–447. Springer, Heidelberg (2004)
[189]. Anand, M., Kim, J., Lee, I.: Code generation from hybrid systems models for distributed embedded systems. In: Proceedings of the IEEE ISORC, pp. 166–173 (2005)
[190]. Anand, M., Fischmeister, S., Kim, J., Lee, I.: Distributed-code generation from hybrid systems models for time-delayed multirate systems. In: EMSOFT 2005: Proceedings of the 5th ACM international conference on Embedded software, pp. 210–213. ACM Press, New York (2005)
[191]. Henzinger, T.A., Ho, P.H.: Algorithmic analysis of nonlinear hybrid systems. In: Wolper, P. (ed.) Proceedings of the 7th International Conference On Computer Aided Verification, Liege, Belgium, vol. 939, pp. 225–238. Springer, Heidelberg
[192]. (1995)
[193]. Agrawal, A., Karsai, G., Ledeczi, A.: An end-to-end domain-driven software development framework. In: OOPSLA 2003: 18th annual ACM SIGPLAN conference
[194]. on OOP, systems, languages, and applications, pp. 8–15. ACM Press, New York
[195]. (2003)
[196]. Chen, K., Sztipanovits, J., Abdelwahed, S., Jackson, E.K.: Semantic Anchoring with Model Transformations. In: ECMDA-FA, pp. 115–129 (2005)
[197]. Chen, K., Sztipanovits, J., Neema, S.: Toward a semantic anchoring infrastructure for domain-specific modeling languages. In: EMSOFT 2005: Proceedings of the 5th ACM international conference on Embedded software, pp. 35–43. ACM Press, New York (2005)
[198]. Denney, E., Fischer, B.: Certifiable Program Generation. In: GPCE, pp. 17–28 (2005)
[199]. G¨ottler, H.: Attributed graph grammars for graphics. In: Proceedings of the 2nd International Workshop on Graph-Grammars and Their Application to Computer Science, London, UK, pp. 130–142. Springer, Heidelberg (1983)
[200]. Holzmann, G.J.: The Model Checker SPIN. Software Engineering 23(5), 279–295 (1997)
[201]. Sch¨auffele, Zurawka (ed.): Automotive Software Engineering. Vieweg & Sohn Verlag, Wiesbaden (2006)
[202]. National Instruments: Web pages of the National Instruments corporation (2007)
[203]. dSpace AG: Web pages of the dSpace corporation (2005)
[204]. Etas Group: Web pages of the Etas Group (2007)
[205]. Vector Informatik GmbH: Web pages of the Vector Informatik GmbH (2007)
[206]. MBtech Group: Web pages of the MBtech Group (2007)
[207]. IEEE: IEEE Std.1800-2005 - Standard for SystemVerilog Unified Hardware Design, Specification and Verification Language (2005)
[208]. IEEE: IEEE Std.1850-2005 - IEEE Standard for Property Specification Language (PSL) (2005)
[209]. IEEE: IEEE Std.1647-2006 - Standard for the Functional Verification Language 'e'(2006)
[210]. Bergeron, J., Cerny, E., Nightingale, A., Hunter, A.: Verification methodology manual for SystemVerilog. Springer, Heidelberg (2006)
[211]. SCC20 ATML Group: IEEE ATML specification drafts and IEEE ATML status reports (2006)
[212]. ETSI: ES 201 873-1 V3.2.1: Methods for Testing and Specification (MTS). The
[213]. Testing and Test Control Notation Version 3, Part 1: TTCN-3 Core Language (2007)
[214]. Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., Pister, K.: System architecture directions for networked sensors. In: Proceedings of the ninth international conference on Architectural support for programming languages and operating systems, pp. 93–104. ACM Press, New York (2000)
[215]. OSEK Consortium: OSEK/VDX communication specification (2004), http://www.osek-vdx.org
[216]. Schmidt, D.C., Stal, M., Rohnert, H., Buschmann, F.: Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects, New York, vol. 2. Wiley & Sons, Chichester (2000)

[217]. Subramonian, V., Gill, C., Sanchez, C., Sipma, H.B.: Composable timed automata models for real-time embedded systems middleware. Technical Report WUCSE-2005-29, Computer Science and Engineering Department, Washington University in St.Louis (2005)
[218]. Wang, N., Gill, C., Schmidt, D.C., Subramonian, V.: Configuring Real-time Aspects in Component Middleware. In: OTM 2004. LNCS, vol. 3291, pp. 1520–1537. Springer-Verlag, Heidelberg (2004)
[219]. Sharp, D.C., Roll, W.C.: Model-Based Integration of Reusable Component-Based Avionics System. In: Proc. of the Workshop on Model-Driven Embedded Systems in RTAS (2003)
[220]. Object Management Group: The Common Object Request Broker: Architecture and Specification. 3.0.2 edn. (2002)
[221]. Balasubramanian, K., Balasubramanian, J., Parsons, J., Gokhale, A., Schmidt, D.C.: A Platform-Independent Component Modeling Language for Distributed Real-time and Embedded Systems. In: Proceedings of the 11th Real-time Technology and Application Symposium (RTAS 2005), San Francisco, CA, pp. 190–199. IEEE, Los Alamitos (2005)
[222]. Sanchez, C., Sipma, H.B., Subramonian, V., Gill, C., Manna, Z.: Thread Allocation Protocols for Distributed Real-Time and Embedded Systems. In: Wang, F. (ed.) FORTE 2005. LNCS, vol. 3731, pp. 159–173. Springer, Heidelberg (2005)
[223]. Behrmann, G., David, A., Larsen, K.G.: A Tutorial on Uppaal. In: Bernardo, M., Corradini, F. (eds.) SFM-RT 2004. LNCS, vol. 3185, pp. 200–236. Springer, Heidelberg (2004)
[224]. Bozga, M., Graf, S., Ober, I., Ober, I., Sifakis, J.: The IF Toolset. In: Bernardo, M., Corradini, F. (eds.) SFM-RT 2004. LNCS, vol. 3185, pp. 237–267. Springer, Heidelberg (2004)
[225]. Robby, Dwyer, M., Hatcliff, J.: Bogor: An Extensible and Highly-Modular Model Checking Framework. In: In the Proceedings of the Fourth Joint Meeting of the European Software Engineering Conference and ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE 2003), Helsinki, Finland, ACM, New York (2003)
[226]. Holtzman, G.J.: The Model Checker SPIN. IEEE Transactions on Software Engineering 23(5), 279–295 (1997)
[227]. Bozga, M., Graf, S., Ober, I., Mounier, L.: IF-2.0: A Validation Environment for ComponentBased Real-Time Systems. Brinksma, E., Larsen, K.G. (eds.) CAV 2002. LNCS, vol. 2404, Springer, Heidelberg (2002).