Digital Twinning Based Adaptive Development Environment for Automotive Cyber-Physical Systems

Guoqi Xie, *Senior Member, IEEE*, Kehua Yang, Cheng Xu, Renfa Li, *Senior Member, IEEE*, and Shiyan Hu, *Senior Member, IEEE*

***Abstract*—Automotive cyber-physical systems need to be rigorously checked and tested under various physical conditions. Automakers aim to improve development efficiency of the automotive cyber-physical systems in the fierce market competition. However, the actual development process suffers from the challenges of long development cycle and poor scalability. To tackle these challenges, this article develops a digital twinning based adaptive development environment for automotive cyber-physical systems, which addresses two critical problems: each physical entity (i.e., electronic control unit, component, test source, etc.) needs to clone a corresponding digital twin; digital twins and the physical entities need to interact closely. The first problem is addressed through proposing an integrated digital twinning clone flow. The second problem is addressed through developing a smart digital twinning board. Our case study with the automotive body control system demonstrates that the adaptive development environment achieves a high adaptability with short development cycle, low complexity, low cost, high scalability, and high flexibility, which meet various automotive cyber-physical design requirements during the development process.**

***Index Terms*—Automotive cyber-physical systems, development environment, digital twinning.**

Manuscript received August 30, 2020; revised December 7, 2020; accepted March 3, 2021. Date of publication March 8, 2021; date of current version October 27, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61932010, Grant 61972139, Grant 61772185, Grant 61702172, and Grant 61672217, in part by the Open Research Project of the Electronic Information and Control of Fujian University Engineering Research Center, in part by Minjiang University, China, under Grant MJXY-KF-EIC1902, and in part by the Fundamental Research Funds for the Central Universities, Hunan University, China. Paper no. TII-20-4131.
(Corresponding authors: Kehua Yang; Cheng Xu; Renfa Li.)

Guoqi Xie, Kehua Yang, Cheng Xu, and Renfa Li are with the Key Laboratory for Embedded and Cyber-Physical Systems of Hunan Province, College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: xgqman@hnu.edu.cn; khyang@hnu.edu.cn; chengxu@hnu.edu.cn; lirenfa@hnu.edu.cn).

Shiyan Hu is with the School of Electronics and Computer Science, University of Southampton, SO17 1BJ Southampton, U.K. (e-mail: s.hu@soton.ac.uk).
Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2021.3064364.

Digital Object Identifier 10.1109/TII.2021.3064364

I. Introduction

A. Motivation

THE AUTOMOBILE industry is a mature industry that has developed over a century. However, to respond to global market competition, low energy and friendly environment, and people's higher requirements for driving, safety, comfort, and autonomy, the automobile industry is facing the great changes of connected, autonomous, shared, and electric [1]. Automotive systems are typical safety-critical cyber-physical systems [2], which needs to run reliably in different driving environments [3]. Cyber-physical systems require interaction and integration between cyberspace and physical entity [3]–[6]. Automotive cyber-physical systems must be strictly checked and tested under physical conditions. If the developed system fails, it will affect the normal use of the automobiles, and might even cause fatal injuries and property damage [3], [7].

As automobiles are mass consumer products, market competition among automakers is fierce. In the development process for automotive cyber-physical systems, automakers aim to reduce development time, reduce workload, improve system quality, and use more efficient ways to control code units, so as to improve development efficiency. However, the actual development process suffers from the following problems.

1) Long development cycle. When developing an automotive cyber-physical system, it is necessary to select different types of products (including driver chips, components, and operating systems) and develop electronic control unit (ECU) control programs. Each set of selection means that a complete product set is needed to design the system. This solution needs long development cycle and high development cost to verify the feasibility and availability of the products.
2) Poor scalability. Once an automotive cyber-physical system is developed, only specific components can often be used. If new functions are required for the automotive cyber-physical system (or new components are introduced), then the existing ECU control program shall be expanded and upgraded (or the ECU and ECU control program shall be redesigned). Such poorly scalable activities often make development quite difficult and cause waste of resources.

B. Contributions

To tackle these challenges, this article develops a digital twinning based adaptive development environment for automotive cyber-physical systems, which addresses two critical problems: each physical entity (i.e., ECU, component, test source, etc.) needs to clone a corresponding digital twin; digital twins and the physical entities need to interact closely. The contributions are as follows.

1) To implement that each physical entity must clone a corresponding digital twin, we propose an integrated digital twinning clone flow.
2) To implement that digital twins and the physical entities must be mixed together interactively, we develop a smart digital twinning board.
3) The case study with the automotive body control system demonstrates that the adaptive development environment achieves a high adaptability with short development cycle, low complexity, low cost, high scalability, and high flexibility, which meet various automotive cyber-physical design requirements during the development process.

II. Related Work

A digital twin is defined as a digital replica of a physical entity [8], [9]. In a digital twin, the information can be transmitted seamlessly such that the replica can behave in the same way as the physical entity. Digital twinning has been applied to many industries, including multimedia technologies [10], composition of power system [11], and secure industrial automation and control system architecture [12], etc.

Digital twinning has been discussed in the field of automotive cyber-physical system testing. Leathrum *et al.* [13] built a test and evaluation (T&E) laboratory for autonomous vehicles. Considering that autonomous vehicles exhibit emergent behavior, they further developed a T&E framework for automotive cyber-physical systems [14]. Through the virtuality-reality spectrum, the testing can migrate between a fully digital environment and a fully physical environment to obtain excellent results [15]. Considering that the simulation-based T&E can reduce the need for physical hardware, Gonda developed a reusable T&E framework, through which testing activities are performed at different levels [16]. The aforementioned studies all adopted the virtual-reality spectrum for the test problem of automotive cyber-physical systems using simulated environments, rather than the digital twinning. Furthermore, they mainly focused on the simulation and test rather than the development. When adopting digital twinning, we need to consider almost all development activities, such as objects, test sources, and test objects, etc., which impose significant challenges.

III. Integrated Digital Twinning Clone Flow

A. Twins in Development Environment

Our development environment consists of two basic twins: first, *ECU twins* (including the physical ECU entity and the digital ECU twin); second, *component twins* (including the physical component entity and the digital component twin).

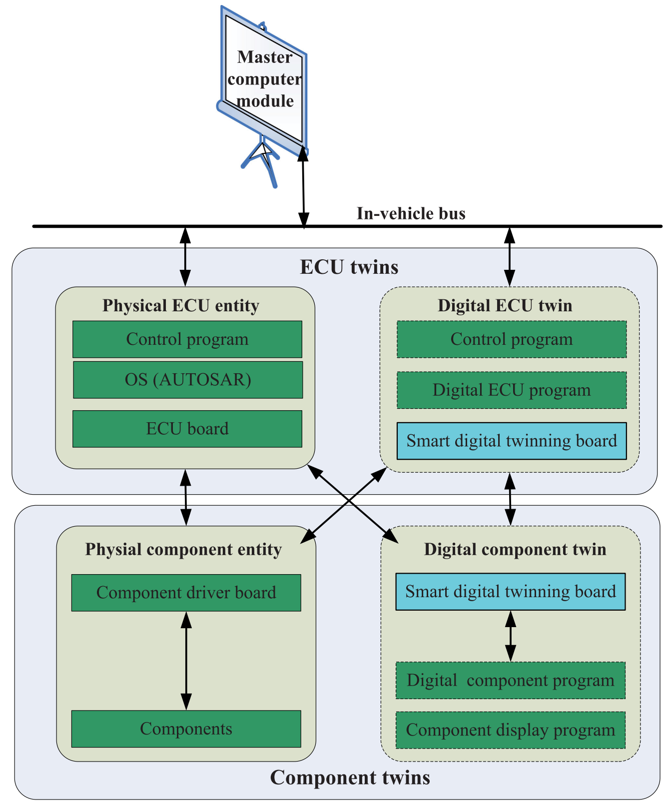

Fig. 1. Two basic twins in development environment for automotive cyber-physical systems.

The development environment organization is shown in Fig. 1. Among them, the physical ECU entity and physical component entity belong to the physical twins, whereas the digital ECU twin and the digital component twin belong to the digital twins.

The master computer module is auxiliary module, which can be used to develop additional functions (e.g., the digital stimulus twin for the body control system). The master computer module collects and records the operation and response of the system through a USB-to-vehicle bus (e.g., USB-to-CAN), and generates corresponding logs to analyze the test results.

B. Integrated Digital Twinning Clone Flow for the Digital ECU Twin

1) Physical ECU Entity: The physical ECU entity mainly consists of ECU board, real-time operating systems based on the AUTOSAR standard, and control program. For an automotive cyber-physical system, corresponding control program is loaded in the physical ECU entity to send control signal to controlled components. For instance, engine ECU, body control ECU, adaptive cruise control (ACC) ECU, brake-by-wire ECU, and airbag ECU have individual control programs. The control program running on the physical ECU entity can be replaced, and developers can determine whether to use the RTOS operating system from different manufacturers.

2) Digital ECU Twin: To ensure that the digital ECU twin has the control ability, it should consist of three parts: digital ECU program, control program, and the digital-twins board.

1) The digital ECU program aims to have the same role as the physical ECU board.
2) The control program in the digital ECU twin is downloaded from the physical ECU entity. By downloading the control program of the physical ECU to the digital ECU program, the digital ECU program can use the same control signals as the physical ECU.
3) Through the smart digital twinning board, digital ECU twin can control the physical component entity. Details of how to achieve the digital-twins board will be discussed in Section IV.

To ensure that the physical ECU entity has an equal digital ECU twin, an integrated digital twinning clone flow for the digital ECU twin is proposed. The detailed steps of the integrated digital twinning clone flow for the digital ECU twin are as follows.

1) By analyzing the chip characteristics of the physical ECU board for the component, the SkyEye simulator running on Linux is used to implement the digital ECU twin.
2) The SkyEye simulator is modified to ensure that the digital ECU program has the same processing speed and communication functionalities [including controller area network (CAN) communication, serial peripheral interface (SPI) communication, analog-to-digital (A/D) conversion, and digital-to-analog (D/A) conversion] as the physical ECU board.
3) We download the control program in the physical ECU entity to the digital ECU program; then, the digital ECU program controls the physical components by connecting the smart digital twinning board.
4) The digital ECU program is tested and modified until the physical component entity can respond normally and its function is completely covered.

The integrated digital twinning clone flow for the digital ECU twin is shown in Fig. 2 .

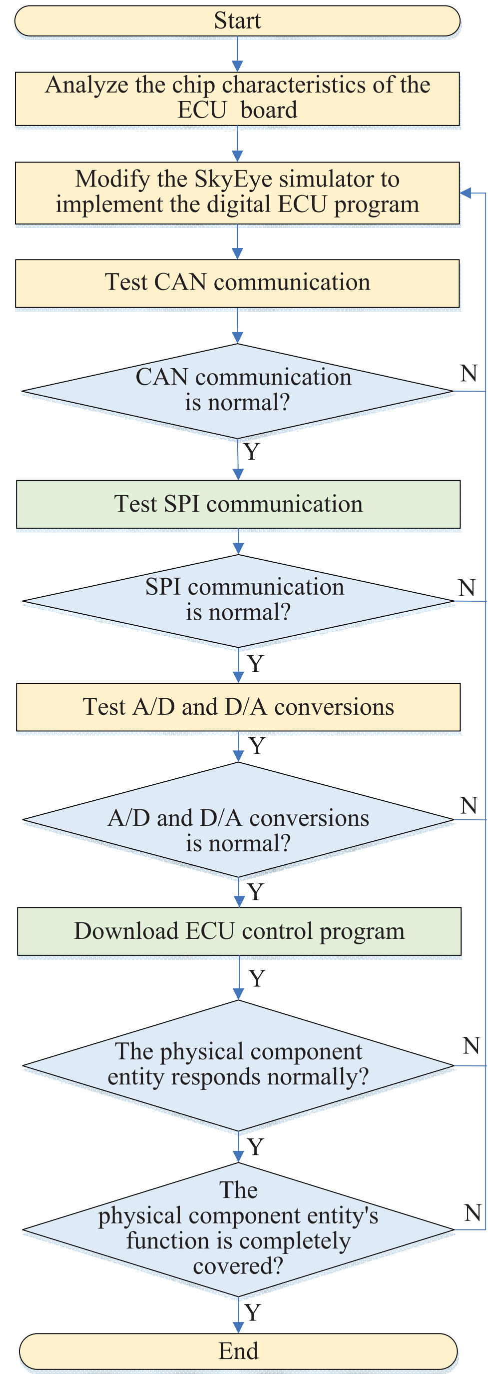

Fig. 2. Integrated digital twinning clone flow for the digital ECU twin.

C. Integrated Digital Twinning Clone Flow for the Digital Component Twin

1) Physical Component Entity: The physical component entity consists of several controlled components (e.g., the body control entity has light, door, seat, wiper components, etc.) and a component driver board (integrated with the driver chips of each components); The component driver board controls the components in the physical component entity.

The ECU board (in the physical ECU entity) and the component driver board (in the physical component entity) are connected to each other through programming input/output model (PIO), such that the physical ECU entity can control the physical component entity, as shown in Fig. 3.

2) Digital Component Twin: To ensure that the digital component twin has the function, it should consist of three parts: digital component program, component display program, and the digital-twins board.

1) The digital component program aims to have the same role as the physical components. By analyzing the characteristics of physical components, Java or Flash are used to write corresponding digital component programs.
2) The component display program in the digital ECU twin shows the result of digital component programs, thereby conveniently and intuitively demonstrating the behavior of digital component programs.
3) Through the smart digital twinning board, ECU twins (including physical ECU entity and digital ECU twin) can control the digital component twin. Details of how to achieve the digital-twins board will be discussed in Section IV.

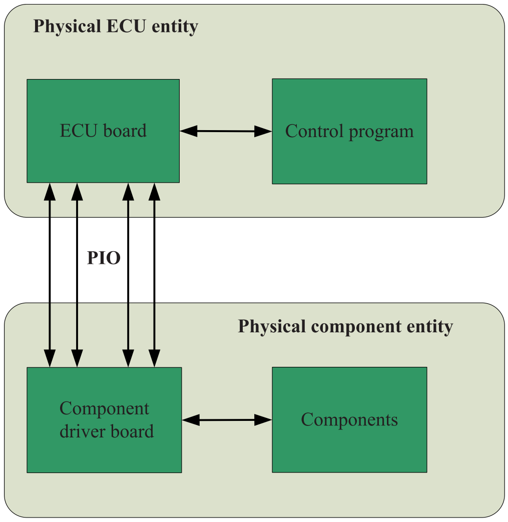

Fig. 3. Physical ECU entity controls the physical component entity.

To ensure that the physical component entity has an equal digital component twin, an integrated digital twinning clone flow for the digital component twin is proposed. The detailed steps of the integrated digital twinning clone flow for the digital component twin are as follows.

1) The driver chip of each component in the physical component entity is analyzed; then, we write driver programs for components and download them to the smart digital twinning board; the smart digital twinning board uses the driver programs to map the functions of the corresponding driver chips.
2) The driver programs are tested and modified until the physical component entity can respond normally and its function is completely covered.
3) We analyze each component of the physical component entity, and write the digital component program to run on the PC; the digital component program accepts the control of the smart digital twinning board, and feedbacks the signal.
4) The digital component programs are tested and modified until digital component twin can respond normally and its function is completely covered.

The integrated digital twinning clone flow for the digital component twin is shown in Fig. 4.

By the integrated digital twinning clone flow, both ECU and component have physical and digital implementations; the digital and physical implementations of each ECU (or component) can coexist, or we can choose one of the two to exist; both digital and physical implementations can fully reflect the functionality of the system, thereby providing developers with intuitiveness and convenience.

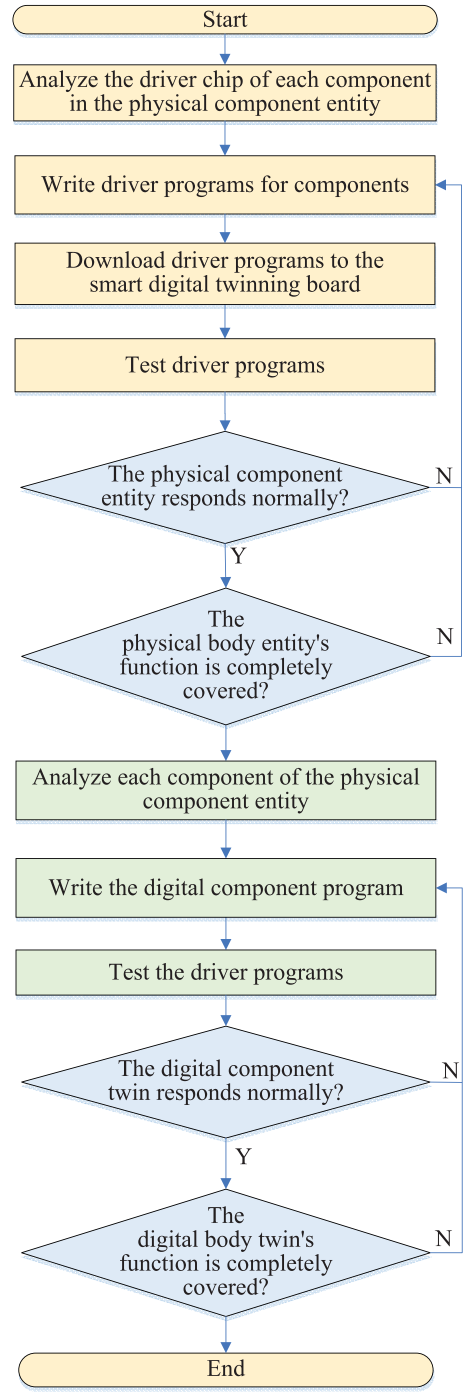

Fig. 4. Integrated digital twinning clone flow for the digital component twin.

IV. Smart Digital Twinning Board

A. Overview of the Smart Digital Twinning Board

To implement that digital twinning and the physical entities are mixed together interactively, we develop a smart digital twinning board, through which both the physical ECU and the digital ECU can control the physical or digital components, and their control signals are consistent. Therefore, the smart digital twinning board is the core of the development environment, and it acts as a route between "digital twin" and "physical entity." The smart digital twinning board is smart because it has two functions, which are as follows.

1) Through PIO, a smart digital twinning board can be connected to the component driver board and the physical

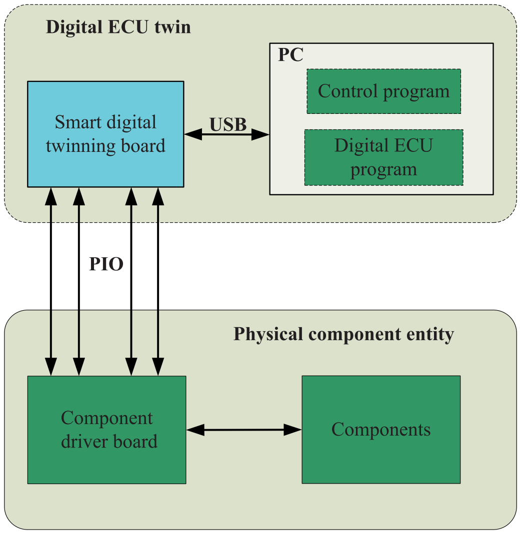

Fig. 5. Digital ECU twin controls the physical component entity in the development environment.

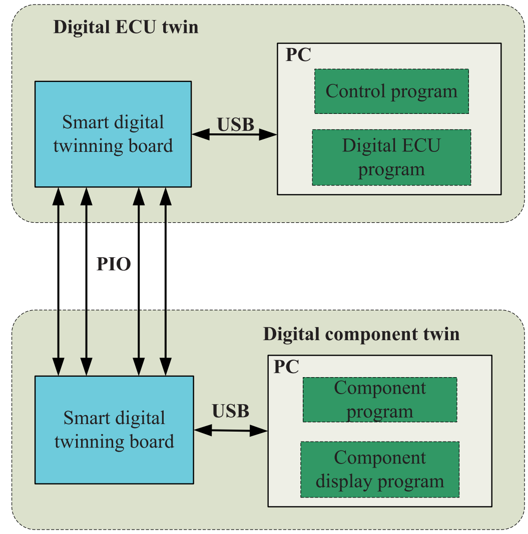

Fig. 6. Digital ECU twin controls the digital component twin in the development environment.

ECU board, as well as another smart digital twinning board.

2) Through analyzing the driver chip of each physical component, the corresponding component's driver program can be written in C language; then, all components' driver programs are downloaded to the smart digital twinning board to realize the mapping between the components driver programs (in the smart digital twinning board) and their corresponding physical driver chips.

In general, it is enough that the smart digital twinning board is implemented by a single-chip microcomputer. In Section V, the body control system development is used as the case study, where the smart digital twinning board is implemented by the C8051F410 single-chip microcomputer.

B. Smart Digital Twinning Board in the Digital ECU Twin

The digital ECU twin can provide the same external interface as the physical ECU entity through the smart digital twinning board. The smart digital twinning board (in the digital ECU twin) and the drive board (in the physical component entity) are connected with each other through PIO, as shown in Fig. 5. Through the routing of the smart digital twinning board in Fig. 5, the digital ECU twin can use the same control signals as the physical ECU entity, such that the digital ECU twin can control the physical components. In this way, the physical components can be controlled by PC software program (i.e., the digital ECU twin), which is convenient for testing and verification and can guarantee that the test results are authentic.

The digital ECU twin can control not only the physical components, but also the digital components, as shown in Fig. 6. Based on Fig. 6, a complete testing and verification system can be established, and the system can still be authentic.

The digital ECU twin has the following advantages.

1) The control programs based on different algorithms can be implemented on the digital ECU twin. For example, for the body control system, developers can use multiple algorithms to prevent window jamming and finally find the best algorithm.

2) Developers can easily and quickly modify the design of the digital ECU program. Compared with the physical ECU board, the digital ECU program does not need to remake the ECU circuit board, thereby saving time and cost.

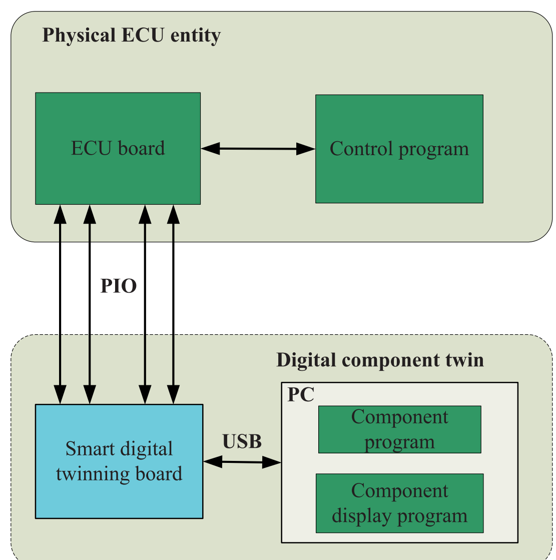

Fig. 7. Physical ECU entity controls the digital component twin in the development environment.

C. Smart Digital Twinning Board in the Digital Component Twin

The digital component twin can be controlled not only by the digital ECU twin (see Fig. 6), but also the physical ECU entity (see Fig. 7). As shown in Fig. 7, through the routing of

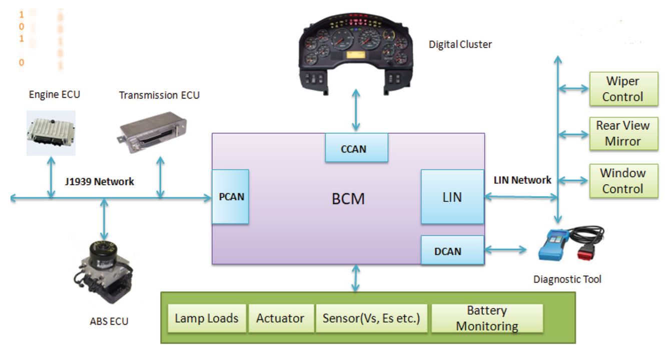

Fig. 8. Body control system in an automobile [17].

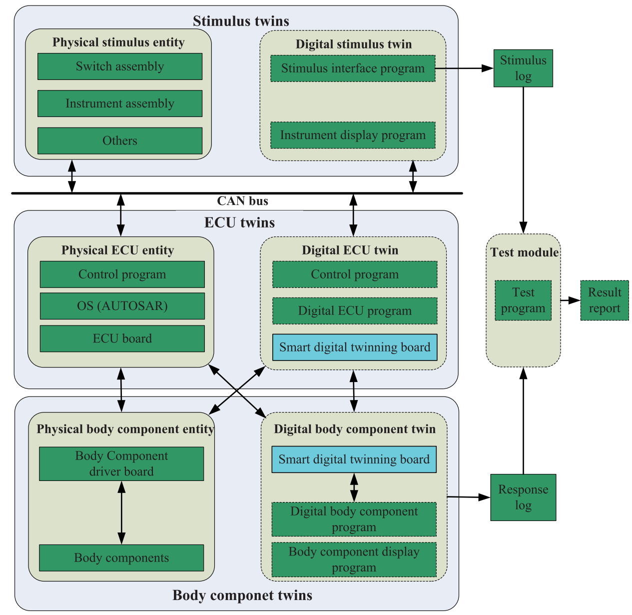

Fig. 9. Twins and module in developing the body control system using the development environment.

the smart digital twinning board, the physical ECU entity can control the digital component twin, such that the control result can be obtained intuitively, and the data can be collected and recorded in the digital component twin.

The digital ECU twin has the following advantages.

1) Through mapping the physical component entity (e.g., the physical lights, doors, windows, and other components in the body component entity) to the digital component twin, it brings a lot of convenience to the actual development, such as a significant reduction in floor space and cost.
2) The digital component program can respond to the control signals from the smart digital twinning board and feedback status and error signals, such as lamp damage or gripping information.
3) The component display program can intuitively display the running status of the components, and record the response log, which is more convenient and faster than the traditional way of manually recording the response log from the physical components.

V. Case Study

A. Body Control System

Body control system, which is mainly named body control module in automotive industry [17], is mainly used to improve the convenience and comfort of automobile driving. The system covers a wide range, including light control, door control, seat control, wiper control, power window control, air conditioning control, and dashboard display, etc. It can be seen that the body control system includes a large number of control objects, as shown in Fig. 8.

B. Twins and Module in the Developing Body Control System

Fig. 9 shows the twins and module in developing the body control system using the adaptive development environment. In addition to the two basic twins (i.e., ECU twins and component twins) provided by the development environment, one additional twins and one additional module are required to be added when developing the body control system: the physical stimulus entity, the digital stimulus twin, and the test module.

1) The physical stimulus entity includes the light switch assembly, the window switch assembly, and the instrument assembly, etc.
2) The digital stimulus twin runs on the host computer module, and it contains the instrument display program and the stimulus interface program. The digital stimulus twin and the physical stimulus entity use the same communication protocol (e.g., the SAE J1939 protocol) to interact with the physical ECU entity or digital ECU twin. The setting steps of the digital stimulus twin are as follows.
   1) We analyze the ECU communication protocol of the body control system, and write the digital stimulus programs (i.e., the instrument display program and the stimulus interface program) of the digital stimulus twin in accordance with the ECU communication protocol.
   2) The digital stimulus programs are tested and modified until the digital component twin can respond normally and its function is completely covered. First, we analyze the ECU communication protocol.
3) The test module is connected to the physical (or digital) stimulus entity (or twin) and the physical (or digital) component entity (or twin) through the local area network. The test program in the test module can verify the test results by comparing the stimulus log with response log, as shown in the right part of Fig. 9.

The reason for virtualizing the stimulus entity is that the digital stimulus twin can repeatedly, orderly, or randomly send control signals and record the stimulus log. It is very inefficient to obtain the test case through manual control of the physical switch assembly in the physical stimulus entity (e.g., 1000 consecutive turns of the lights).

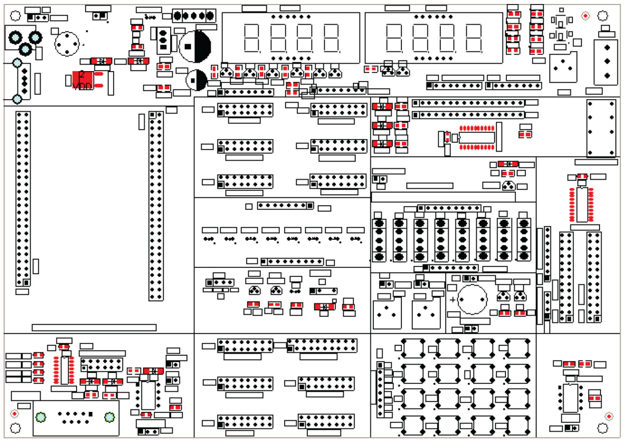

Fig. 10. MC9S12DG128-based body control program motherboard in the physical ECU entity.

C. Software/Hardware Details of Twins

1) The physical ECU entity uses Freescale's 16-b microcontroller MC9S12DG128 [18] as the computing unit of the ECU. This chip uses HCS12 CPU core, 128-KB Flash EEPROM, 8-KB RAM, and 2-KB EEPROM. The ECU circuit board provides SPI bus and general-purpose input/output (GPIO) interfaces. These interfaces are used to connect the physical body component entity. A/D and pulsewidth modulation pins are provided in each interface to collect signals from the driver chip and control body components. Fig. 10 shows the MC9S12DG128-based motherboard of the body control program in the physical ECU entity.
2) The digital ECU twin uses the SkyEye simulator and runs on the Linux operating system. Provided by the smart digital twinning board, the digital ECU twin has the same external interface as the physical ECU entity.
3) The physical body component entity includes lights, doors, and windows, etc. For example, the MC33888 driver chip [19] is used for lights, and it provides a 16-pin PIO interface to connect with the ECU board and 12 ports to control the lights. If we select another driver chip, we just need to replace it with the corresponding component driver board. Fig. 11 shows the circuit diagram of the MC33888 driver chip used in the development of the body control system.
4) The smart digital twinning board in the digital body component twin uses a C8051F410 single-chip microcomputer [20], which provides a 16-pin GPIO interface and a 10-pin GPIO interface; furthermore, C8051F410 provides a USB interface to connect with a PC. Fig. 12 shows the minimal system of the 8051F410 single-chip microcontroller used in the development of the body control system.

Fig. 13 shows the development environment for headlight and taillight components. The conventional power source for vehicle lights is a lead-acid battery with 12 V and 16 Ah. In

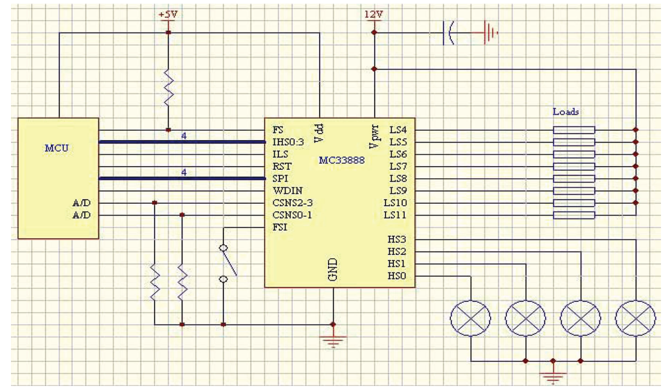

Fig. 11. Circuit diagram of the MC33888 driver chip.

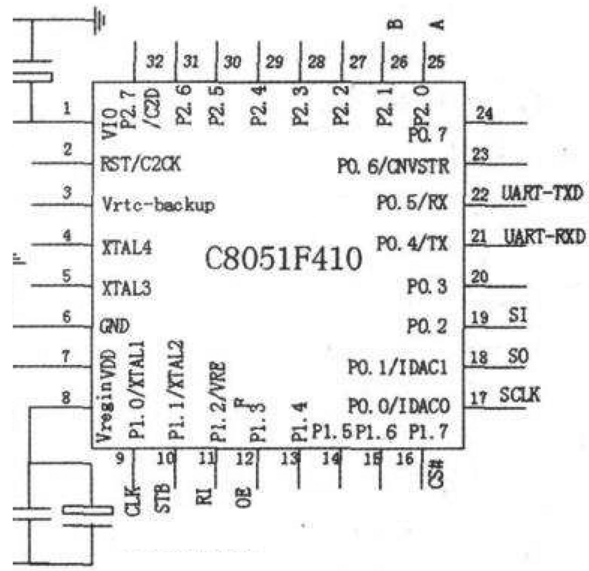

Fig. 12. Minimal system of the C8051F410 single-chip microcontroller.

this development environment, adjustable lead-acid batteries are adopted.

D. Development Process of the Body Control System

The development process for the body control system using the development environment is as follows.

1) We choose the stimulus entity (or twin), ECU entity (or twin), and body component entity (or twin) to build a development environment for the body control system. The digital and physical implementation of each stimulus (or ECU, component) can coexist or only one of the two exists.
2) The stimulus entity (or twin) sends test stimuli to the physical (or digital) ECU entity (or twin). If the stimulus entity (or twin) is the digital stimulus twin, test stimuli are generated in three ways: random, coding, and file import; otherwise, if the stimulus entity (or twin) is the physical stimulus entity, then test stimuli are generated by manual operation (i.e., control switch assembly).

Fig. 13. Development environment for headlight and taillight components using the development environment.

3) If the digital stimulus twin is used, all generated test stimuli are recorded automatically into the stimulus log file; otherwise, if the physical stimulus entity is used, all generated test stimuli are recorded manually into the stimulus log file. In other words, the entire stimulus process using the digital stimulus twin can be automatically recorded, which is convenient for debugging and modification.
4) The stimulus log file is sent to the test module.
5) The physical (or digital) body component entity (or twin) receives test stimuli and controls the physical (or digital) body component entity (or twin) through the body control program. The physical (or digital) body component entity (or twin) receives control and responds. If the digital body component twin is used, all generated responses are recorded automatically into the response log file; otherwise, if the physical body component entity is used, all generated responses are recorded manually into the response log file.
6) The response log file is sent to the test module.
7) The test module compares the response log file with the stimulus log file. If the two log files are consistent, the body control system satisfies the design requirements; otherwise, if there is a deviation, we should analyze the corresponding reason, and need to improve the body control system design by modifying and debugging the corresponding entities.

The flowchart of the aforementioned development process for the body control system using the development environment is shown in Fig. 14.

Finally, the stimulus test is conducted. Within the specified 72 h, the digital stimulus twin sent a total of 17 685 test stimuli to the digital ECU twin; the digital body twin received a total of 17 685 responses to the test stimuli; therefore, the accuracy of the stimulus test is 100%, and the misoperation rate is 0%, as shown in Table I. In other words, the body component can still run normally under the long-term and high-frequency stimulus test. The results verify the stability of the body control system and the effectiveness of the system design.

The digital twinning based development environment in this article is developed for automotive cyber-physical systems. As

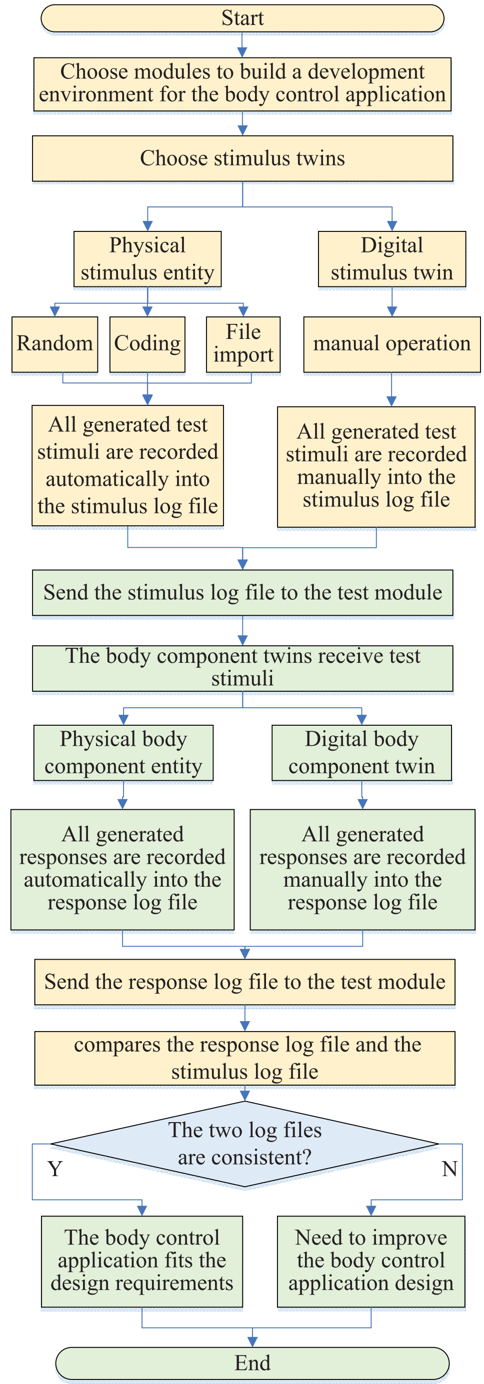

Fig. 14. Development process for the body control system using the development environment.

TABLE I
STIMULUS TEST RESULTS FOR THE BODY CONTROL SYSTEM

Test conditions	Result
Time span	72 hours
Number of command sent	17,685
Number of response	17,685
Accuracy	100%
Misoperation rate	0%

far as we know, this is the first work that digital twinning technology has been used to implement an adaptive development environment. The purpose is to replace some complicated physical inspections and tests with digital twinning technology, thereby improving development efficiency. This article focuses on introducing cases and tests, rather than comparing algorithms and numerical experiments. What we emphasize is that how the development environment is implemented, and the benefits of the development environment in practice.

E. Benefits of the Development Environment

On the basis of the case study with the body control system, the development environment achieves the following high adaptability for automotive cyber-physical system development.

1) Short development cycle. After the development environment virtualizes the physical ECU and physical components, a large number of repeatable and customizable cases can be covered in a programmable mode, thereby improving development efficiency.
2) Low complexity. The development environment can not only easily test the effectiveness of control programs, driver programs, and operating systems in physical ECU entity, but also easily determine component drive chip selection and component selection, thereby quickly building and testing product solutions for an automotive cyber-physical system.
3) Low cost. The development environment makes the digital ECU twin be implemented in the form of SkyEye, which can quickly and easily modify the design without remaking the ECU circuit board, thereby saving time and cost.
4) High scalability. The development environment is convenient for programmers to extend or change system design solutions and product solutions; both digital and physical implementations can fully reflect the functionality of the system, thereby providing developers with intuitiveness and convenience.
5) High flexibility. The development environment can intuitively display the test response, and can automatically save the response record to the log file to achieve unattended automatic testing and statistics.

Overall, the development environment highlights the high adaptability with short development cycle, low complexity, low cost, high scalability, and high flexibility, and can basically meet various design requirements during the development process.

VI. Conclusion

By adopting digital twinning, this article built an adaptive automotive cyber-physical system development environment to improve development efficiency. The development environment was a complete and realistic digital twinning based development environment by sending consistent control signals from the physical (or digital) ECU to the physical (or digital) components. The development environment was a successful realization of the digital twinning solution in automotive cyber-physical system development, and has shown considerable development efficiency. The development environment can help automakers to demonstrate the high adaptability with short development cycle, low complexity, low cost, high scalability, and high flexibility in actual automotive cyber-physical system development.

References

[1] RK&K, “Connected, autonomous, shared, and electric vehicles (CASE),” 2019. [Online]. Available: https://www.rkk.com/our-work/services/innovation/connected-autonomous-shared-and-electric-vehicles/
[2] G. Xie, Y. Li, Y. Han, Y. Xie, G. Zeng, and R. Li, “Recent advances and future trends for automotive functional safety design methodologies,” *IEEE Trans. Ind. Inform.*, vol. 16, no. 9, pp. 5629–5642, Sep. 2020.
[3] G. Xie *et al.*, “Reliability enhancement toward functional safety goal assurance in energy-aware automotive cyber-physical systems,” *IEEE Trans. Ind. Inform.*, vol. 14, no. 12, pp. 5447–5462, Dec. 2018.
[4] X. Wang *et al.*, “A multi-order distributed HOSVD with its incremental computing for big services in cyber-physical-social systems,” *IEEE Trans. Big Data*, vol. 6, no. 4, pp. 666–678, Dec. 2020.
[5] X. Wang, L. T. Yang, Y. Wang, X. Liu, and M. J. Deen, “A distributed tensor-train decomposition method for cyber-physical-social services,” *ACM Trans. Cyber-Phys. Syst.*, vol. 3, no. 4, pp. 1–15, Oct. 2019.
[6] X. Wang, L. T. Yang, X. Chen, J. J. Han, and J. Feng, “A tensor computation and optimization model for cyber-physical-social big data,” *IEEE Trans. Sustain. Comput.*, vol. 4, no. 4, pp. 326–339, Dec. 2019.
[7] L. L. Bello, R. Mariani, S. Mubeen, and S. Saponara, “Recent advances and trends in on-board embedded and networked automotive systems,” *IEEE Trans. Ind. Inform.*, vol. 15, no. 2, pp. 1038–1051, Feb. 2019.
[8] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, “Digital twin in industry: State-of-the-art,” *IEEE Trans. Ind. Inform.*, vol. 15, no. 4, pp. 2405–2415, Apr. 2019.
[9] Y. Fang, C. Peng, P. Lou, Z. Zhou, J. Hu, and J. Yan, “Digital-twin-based job shop scheduling toward smart manufacturing,” *IEEE Trans. Ind. Inform.*, vol. 15, no. 12, pp. 6425–6435, Sep. 2019.
[10] A. El Saddik, “Digital twins: The convergence of multimedia technologies,” *IEEE MultiMedia*, vol. 25, no. 2, pp. 87–92, Apr.–Jun. 2018.
[11] S. K. Andryushkevich, S. P. Kovalyov, and E. Nefedov, “Composition and application of power system digital twins based on ontological modeling,” in *Proc. IEEE 17th Int. Conf. Ind. Inform.*, 2019, pp. 1536–1542.
[12] C. Gehrmann and M. Gunnarsson, “A digital twin based industrial automation and control system security architecture,” *IEEE Trans. Ind. Inform.*, vol. 16, no. 1, pp. 669–680, Jan. 2020.
[13] J. F. Leathrum, R. R. Mielke, Y. Shen, and H. Johnson, “Academic/industry educational lab for simulation-based test & evaluation of autonomous vehicles,” in *Proc. Winter Simul. Conf.*, 2018, pp. 4026–4037.
[14] J. Leathrum, Y. Shen, R. Mielke, and N. Gonda, “Integrating virtual and augmented reality based testing into the development of autonomous vehicles,” in *Proc. ModSim World*, 2018, pp. 24–26.
[15] T. Laverghetta and C. Hall, “Test and evaluation of autonomous vehicles using simulated environments,” 2019. [Online]. Available: https://digitalcommons.odu.edu/undergradsymposium/2019/postersession/10/
[16] N. D. Gonda, “A framework for test & evaluation of autonomous systems along the virtuality-reality spectrum,” M.S. thesis, Dept. Model. Simul. Vis. Eng., Old Dominion Univ., Norfolk, VA, USA, 2019.
[17] E. Technologies, “How body control module (BCM) works in an automobile,” 2020. [Online]. Available: https://www.embitel.com/automotive-insights/understanding-body-control-module-the-brain-of-body-electronics-in-automotive
[18] “MC9S12DG128—NXP Semiconductors,” 2005. [Online]. Available: https://www.nxp.com/docs/en/fact-sheet/MC9S12DG128FS.pdf
[19] “Quad high-side and octal low-side switch for automotive,” 2004. [Online]. Available: https://datasheetspdf.com/pdf-file/541544/Motorola/MC33888/1
[20] “C8051F410/1/2/3—Silicon Labs,” 2004. [Online]. Available: https://www.silabs.com/documents/public/data-sheets/C8051F41x.pdf

Guoqi Xie (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from Hunan University, Changsha, China, in 2014.

He is currently a Professor with Hunan University. He was a Postdoctoral Research Fellow with Nagoya University, Nagoya, Japan. His current research interests include real-time systems, automotive embedded systems, embedded safety, and security.

Dr. Xie was the recipient of the 2018 IEEE TCSC Early Career Researcher Award. He is currently serving on the Editorial Boards of the *Journal of Systems Architecture*, *Microprocessors and Microsystems*, and *Journal of Circuits, Systems and Computers*. He is an ACM Senior Member.

Kehua Yang received the Ph.D. degree in computer science and engineering from Southeast University, Nanjing, China, in 2005. He is currently an Associate Professor with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. His major research interests includes embedded systems, cyber-physical systems, and automotive systems.

Cheng Xu received the Ph.D. degree in mechanical and electrical engineering from the Wuhan University of Technology, Wuhan, China, in 2006. He is currently a Full Professor with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. His major research interests include embedded systems, cyber-physical systems, and automotive systems.

Renfa Li (Senior Member, IEEE) received the Ph.D. degree in electronic engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2002. He is currently the Professor and Chair in embedded and cyber-physical systems with Hunan University, Changsha, China. He is currently the Chair of the Key Laboratory for Embedded and Cyber-Physical Systems. His major research interests include computer architectures, embedded computing systems, cyber-physical systems, and Internet of Things.

Prof. Li is a Member of the council of CCF and a Senior Member of ACM.

Shiyan Hu (Senior Member, IEEE) received the Ph.D. degree in computer engineering from Texas A&M University, College Station, TX, USA, in 2008.

He is currently the Professor and Chair in cyber-physical systems with the University of Southampton, Southampton, U.K. His research interests include cyber-physical systems and cyber-physical system security.

Dr. Hu is currently the Chair for IEEE Technical Committee on Cyber-Physical Systems. He is the Editor-in-Chief for the *IET Cyber-Physical Systems: Theory & Applications*. He is an Associate Editor for the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, *ACM Transactions on Design Automation of Electronic Systems*, and *ACM Transactions on Cyber-Physical Systems*. He has held Chair positions in numerous IEEE/ACM conferences. He is a Fellow of IET and a Fellow of British Computer Society.