Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 09, September - 2025

A Comprehensive Review of Unified Diagnostic Services (UDS) in Automotive Systems: Architecture, Capabilities, Limitations and Future Perspective

Sid Pasumarthi
Automotive Professional and Independent Researcher,
Michigan, USA,

Sivani Manisha Vankayala
Automotive Professional and Independent Researcher,
Michigan, USA,

***Abstract* - Unified Diagnostic Services (UDS) as defined by ISO 14229, has become foundational to automotive diagnostic systems by providing a standardized interface between Electronic Control Units (ECUs) and external development and test equipment. The paper explores the rationale behind UDS by studying the shortcomings of earlier diagnostic systems, such as KWP2000, OBD-II, and ISO 15765, which were far more siloed and lacked robust lifecycle support. The paper further provides a layered architectural view of UDS in terms of application, transport, and network/physical layers, it describes the full range of Service Identifiers (SIDs) with representative case studies in vehicle production, aftersales maintenance, and field installation updates. The paper goes on to examine prominent issues—timing, security weaknesses, and fragmentation within OEMs—and concludes with ways to enhance UDS with more focus given to cybersecurity, seamless connectivity, and seamless over-the-air distribution.**

***Keywords*— Unified Diagnostic Services (UDS), Electronic Control Units (ECU), Service Identifiers (SID)**

I. INTRODUCTION

Modern automobiles have been transformed into complex distributed electronic systems, with a considerable number of Electronic Control Units (ECUs) in each vehicle, which control the specialized domain functionalities such as Powertrain Control, Active Safety, Infotainment System, and Advanced Driver Assistance Systems (ADAS) among others. Increased in the number of subsystems and their complexities raised, the diagnostic capabilities can no longer apply only during production testing but are equally important for aftersales service making the necessity to be serviceable throughout the life of the vehicle.

Unified Diagnostic Services (UDS) protocol standardized by ISO 14229 through a comprehensive set of diagnostics, documentation of potential services, and a path on how they could be implemented, all serve to support this need. UDS describes a collection of diagnostic communications services for every part of the vehicle life cycle including ECU Development, End of Line (EOL) testing, workshop maintenance and in field software updates. By establishing a consistent interface between diagnostic tools and ECUs, UDS enhances interoperability between diagnostics tools and the vast array of ECUs, decreases service complexity, and facilitates scalable diagnostic support for an ever more complex and diverse range of vehicles and vehicle architectures.

Its relevancy has never been more pronounced than as the industry pivots towards electrification, connectivity, and automated driving, which demand diagnostics that are more robust and provide safety and maintainability assurances. UDS is not only the underpinning of today's diagnostics but also a facilitator of future opportunities

II. BACKGROUND AND MOTIVATION OF DEVELOPING UNIFIED DIAGNOSTIC SERVICES (UDS)

In the last thirty years, automotive diagnostics have changed dramatically. A combination of different standards and proprietary systems existed prior to UDS and that was often inefficient for both development and service. The framework closest to being a common standard was OBD-II (On-Board Diagnostics, ISO 15031/SAE J1979) developed in the 1990s to monitor functions related to emissions and allow access to standard fault codes. While OBD-II was effective for regulatory reasons, it was limited in nature and lacked coverage for the increasingly complicated software-based systems in automobiles.

To provide a broader diagnostic service, Keyword Protocol 2000 (KWP2000, ISO 14230) is proposed which includes more advanced diagnostics over K-Line and CAN. In addition, ISO 15765 defined transport services but did not provide an application layer. OEMs made things worse by creating proprietary extensions that were also for internal reasons producing a fragmented environment where tools, testers and ECUs rarely worked across manufacturers.

Increased functioning complexity of ECUs and the increasing need of software flashing, parameterization and lifecycle management increased the need for a single diagnostic protocol. In 2006, the International Organization for Standardization (ISO) released Unified Diagnostic Services (UDS, ISO 14229), covering and extending diagnostic communications into one harmonized framework. UDS supports a complete service package that covers fault analysis, data access, ECU reprogramming, and communication control, thereby addressing the limitations of earlier standards and ensuring compatibility across the automotive industry.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 09, September - 2025

Timeline showing the Evolution of Automotive Diagnostics

- 1996 – OBD-II (ISO 15031/SAE J1979): Introduced for emissions-related diagnostics, providing access to standardized fault codes.
- 2000 – KWP2000 (ISO 14230): Expanded services, initially over K-Line and later adapted to CAN.
- 2004 – ISO 15765 (Diagnostics on CAN): Defined diagnostic transport layer over CAN but did not unify services.
- 2006 – UDS (ISO 14229): Introduced as a comprehensive and unified protocol, covering data access, reprogramming, and DTC management across the ECU lifecycle.

III. UDS ARCHITECTURE

The Unified Diagnostic Services (UDS) architecture is specified in ISO 14229 and is designed based on a layered abstraction that encapsulates diagnostic services from specific transport and physical communication layers. This allows UDS to be a standalone functional and represent evolution of diagnostics while being compatible with legacy systems, and new vehicle network technologies.

A. Layered Model

The Application Layer (ISO 14229-1) defines diagnostic services, manages sessions, and encapsulates Diagnostic Trouble Codes (DTC). Each diagnostic function is denoted by a Service Identifier (SID) and may have optional sub-function definitions.

The Transport Layer packages the diagnostic messages back-to-back and enables messages that exceed the payload limit to the underlying network. For CAN, ISO 15765-2 addresses the ISO-TP protocol that supports splitting the payload and reconstructing diagnostic messages by way of Single Frames, First Frames, Consecutive Frames, and Flow Control frames. For Ethernet applications, the ISO 13400 (DoIP) protocol uses TCP/IP for the higher throughput capabilities while ensuring reliable data transfer to support bulk downloads of software images as compared to other legacy methods.

The Network and Physical Layers specifies how UDS payloads are encapsulated over the specific buses. The CAN specification included identifiers such as 0x7E0 (tester request) and 0x7E8 (ECU response). FlexRay also supports pre-defined timing for the real-time domains, and Ethernet supports the scalability and performance needed for connect and autonomous vehicle platforms.

B. Client-Server Paradigm

UDS communication follows the client-server paradigm, when the external tester (client) will send requests and the ECU (server) it will respond. Depending on the architecture, a UDS request may also traverse several communication gateways that receive and pass diagnostic requests between different in-vehicle networks. Although the UDS protocol enables a client to send requests to single or multiple ECUs, there are two addressing modes that are defined in UDS which will be described in the later section of UDS architecture and its overview.

C. Addressing Modes

UDS standard details two addressing modes: physical addressing, which addresses a specific ECU, and functional addressing, which addresses all ECUs and allows those ECUs to respond as relevant. This flexibility allows the UDS protocol to be used in larger more complex distributed architectures with many controllers.

D. Time Model

UDS, timing requirements are especially stringent around P2 (which specifies a maximum ECU response time) and P2* (which specifies an extended response time when processing is delayed). The transport layer adds flow control timers to respect the timing of message segmentation. The importance of these timing rules is underscored by the need for interoperability of different OEM testers and supplier ECUs.

E. Error Response

Negative Response Codes (NRCs) have largely been standardized so that services around errors would be uniformly handled among test devices. Some NRC situations include ServiceNotSupported (0x11), ConditionsNotCorrect (0x22) and ResponsePending (0x78). By following this framework, it is much easier for testers and other devices to recognize the point of failure, whether on the communication layer or the execution of service request.

F. Security and Session Constraints

Depending on session control (default, extended, program, or OEM) the service requested may also require security access via the seed–key challenge (SID 0x27). There are also various environmental limitations on what services can or cannot be accessed, for example ignition on/off, vehicle battery voltage level or vehicle speed. All of the above serve to establish an appropriate level of safety or compliance.

G. AUTOSAR Realization

In a standard AUTOSAR ECU, the UDS services are separated into different modules, but all are still persistent in the ECU. The Diagnostic Communication Manager (DCM) interprets the diagnostic message and manages service execution. The Diagnostic Event Manager (DEM) stores and reports DTCs. The Communication Stack (ComStack) implements the transport layer of ISO-TP. The Non-volatile Memory Manager (NVM) is used to ensure diagnostic data that persists over time due to the issues associated with memory management. Such a set, enables diagnostics across all suppliers and allows for interoperability and reusability.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 09, September - 2025

Layer	Standards	Functionality
Application Layer	ISO 14229-1 (UDS)	Defines services, sessions, DTC handling, error codes.
Transport Layer	ISO 15765-2 (CAN-TP), ISO 13400	Segmentation, flow control, reliable transfer over CAN/Ethernet
Network Layer	ISO 15765-4 (CAN), ISO 10681 (FlexRay), ISO 9141/14230 (K-Line)	Diagnostic mapping to specific networks.
Physical Layer	CAN, FlexRay, Ethernet, K-Line	Bit-level transmission and electrical signaling.

Table I: UDS Layers and Standards

IV. UDS CAPABILITIES

Unified Diagnostic Services (UDS) provides comprehensive suite of diagnostic and maintenance services for ECUs throughout the vehicle lifecycle from production line tests through fixing problems in the aftersales phase, and reprogramming in the field. The capability of UDS is organized into sessions, and Service Identifiers (SIDs).

A. Session and sub-sessions:

UDS services are governed by the sessions each indicates an operating mode of the ECU relating to the availability of service:

a. Default Session: Basic diagnostics like reading identifiers or monitoring emission related diagnostics.

b. Extended Session: Provides a more extensive scope of operation to allow advanced testing, actuator testing, and data manipulation.

c. Programming Session: Reserved for memory operations and ECU flash programming.

d. OEM-Specific Sessions: Proprietary mode of dealer defined relative to the OEM.

There are sub-sessions or conditions within each of the sessions as it defines accessibility. For example, reprogramming services may only be available in the context of a programming sub-session, with valid security access, and provided satisfactory external conditions (e.g., ignition ON, stable voltage).

B. Service Identifiers (SIDs):

The SID relates to each diagnostic operation. The SID is a one-byte command that tells the ECU which service is being requested. The SID can have sub-function parameters that describe what actions need to be taken for the service. For example, the SID 0x10 (Diagnostic Session Control) can send a request-to-request Default (0x01), Extended (0x03), or Programming (0x02) sessions. The ECU will respond with the accepted/denied/pending status.

C. Core Capabilities and Associated Services:

UDS equips ECUs with a wide spectrum of functions that cover the entire diagnostic lifecycle.

a. Session and Mode Management*:* Defines availability of services via Default, Extended, Programming, or OEM sessions. Session control is handled by SID 0x10, with Tester Present (0x3E) maintaining active sessions.

b. ECU Reset and Power Management: SID 0x11 provides hard, soft, and key-cycle resets for reinitialization, commonly used after programming.

c. Security and Access Control: SID 0x27 secures critical services through a seed–key challenge. It restricts sensitive operations like flashing and actuator overrides.

d. Data Reading and Monitoring: Services such as 0x22 (Read DID), 0x23 (Read Memory), 0x24 (Scaling Data), 0x29 (Read by Address), and 0x2A (Periodic Data) support data retrieval for service and testing.

e. Data Writing and Control: Services 0x2E (Write DID), 0x2F (I/O Control), and 0x2C (Dynamic DID) enable parameter updates and actuator overrides.

f. Routine Execution: SID 0x31 allows execution of internal ECU routines such as self-tests and regeneration.

g. Diagnostic Trouble Code (DTC) Handling: SIDs 0x19, 0x14, and 0x85 provide reading, clearing, and control of Diagnostic Trouble Codes.

h. Programming and Software Download: Services 0x34, 0x35, 0x36, 0x37, and 0x38 enable secure ECU reprogramming.

i. Communication Management: SID 0x28 controls communication channels (enable/disable Rx/Tx).

j. Timing and Communication Parameters: SIDs 0x83, 0x84, 0x86, and 0x87 manage diagnostic timing, security, event-driven responses, and link state.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 09, September - 2025

Function Domain	Key Services (SIDs)	Example Use Case
Session Management	0x10, 0x3E	Switch to Extended Session, keep alive
ECU Reset	0x11	Restart ECU after programming
Security Access	0x27	Unlock flashing operations
Data Reading	0x22, 0x23, 0x24, 0x29, 0x2A	Read VIN, memory values, live data
Data Writing/Control	0x2E, 0x2F, 0x2C	Modify coding, override actuators
Routine Execution	0x31	Run self-test, DPF regeneration
DTC Handling	0x19, 0x14, 0x85	Retrieve and clear fault codes
Programming	0x34, 0x35, 0x36, 0x37, 0x38	ECU re-flash, calibration update
Communication Control	0x28	Suppress CAN traffic
Timing/Comms Parameters	0x83, 0x84, 0x86, 0x87	Adjust timers, event triggers

Table 2: Summary of UDS Functional Domains and Services

V. CHALLENGES TO UDS

Unified Diagnostic Services (UDS) serves as a backbone to the automotive diagnostics and its adoption has significant challenges.

Firstly, the key limitation is security: most common seed–key solution (SID 0x27) provides only a low level of security and is still vulnerable to brute-force and replay attacks, meaning that sensitive functions such as flashing ECU firmware does not have adequate protection.

Second challenge is timing, in particular the P2 or P2* timing must be strictly adhered to and with difference in variations of timing would lead to interoperability issues.

The next challenge comes with the fragmentation when UDS is applied to various OEMs; all OEMs are likely to customize the way they handle sessions, the identifiers they use, and the service flow of the interface. This could lead to a possible interoperability experience despite the availability of standardized UDS. The environmental conditions under which services can be executed, such as the ignition state, unstable voltage, or vehicle movement, will create an additional challenge to achieve a real-world experience.

Finally, the adoption of Diagnostics Over IP (DoIP) creates risk for cybersecurity protection, simply put, the surface area for attack increases in a connected vehicle environment.

Collectively, the challenges to adopt UDS show the tensions between standardization, security, and adaptability that exist in the evolving automotive ecosystem.

VI. FUTURE SCOPE

The growth of Unified Diagnostic Services (UDS) will be determined by increasing demands for cybersecurity, connectivity, and automation. UDS will no longer use legacy seed–key mechanisms that simply relied on cryptographic authentication techniques for securing critical operations; they will upgrade to stronger cryptographic authentication aligned with ISO 21434. UDS will integrate further with the over-the-air (OTA) updates that expand the role of UDS in connected ecosystems to facilitate remote reprogramming and diagnostics. Additionally, self-optimizing adaptive diagnostics that could maximize timing and data gathering variables will improve robustness. By creating harmonization of the data identifiers and session behaviors among the OEMs, UDS can reduce fragmentation of a diagnostic framework, allowing it to eventually be interoperable and scalable enough for tomorrow's electric and autonomous vehicles.

VII. CONCLUSION

Unified Diagnostic Services (UDS) has placed itself firmly in the automotive diagnostic space, uniting communication between Electronic Control Units (ECUs) and external test equipment, across the lifecycle of the vehicle itself. Its layered architecture, session control, and service identifiers provide powerful but consistent measures of diagnosis capability. UDS will continue to move forward within the automotive diagnostic communications space, as it combats no shortfall of challenges in terms of security, strictness in timing, OEM differentiators and fragmentation. UDS will adapt and even innovate as it incorporates improved security, harmonization, and integrations with all things connected, helping to keep UDS in good standing in an era of electric, autonomous, software defined vehicles while still remaining viable, consistent, and relevant for all players that regard safety as paramount in a new world of automotive systems.

VIII. REFERENCES

[1] ISO 14229-1: Road Vehicles – Unified Diagnostic Services (UDS).
[2] ISO 15765-2: Road Vehicles – Diagnostics on CAN.
[3] ISO 13400: Road Vehicles – Diagnostics over IP (DoIP).
[4] J. Schäuffele and T. Zurawka, Automotive Software Engineering, Springer, 2016.
[5] M. Stallmann, “Secure On-Board Diagnostics for Connected Vehicles,” IEEE Transactions on Vehicular Technology, 2020.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 09, September - 2025

IX. APPENDIX – UDS SERVICES AND SUB-SERVICES

SID	Service Name	Example Sub-Functions / Notes
0x10	Diagnostic Session Control	0x01 Default; 0x02 Programming; 0x03 Extended; OEM-specific sessions.
0x11	ECU Reset	0x01 Hard Reset; 0x02 Key Off–On Reset; 0x03 Soft Reset.
0x14	Clear Diagnostic Information	Clears DTC memory.
0x19	Read DTC Information	0x02 Report DTC by Status Mask; 0x04 Report DTC Snapshot; 0x14 Clear DTC.
0x22	Read Data by Identifier (DID)	Access static data such as VIN, software version, calibration ID.
0x23	Read Memory by Address	Raw memory access (restricted to programming sessions).
0x24	Read Scaling Data by Identifier	Retrieve scaling/units for data interpretation.
0x27	Security Access	0x01 Request Seed; 0x02 Send Key.
0x28	Communication Control	Enable/disable Rx/Tx or specific communication channels.
0x29	Read Data by Address	Extended memory read operations.
0x2A	Read Periodic Data	Continuous data streaming.
0x2C	Dynamically Define Data Identifier	Combine multiple DIDs into custom datasets.
0x2E	Write Data by Identifier (DID)	Parameter reconfiguration.
0x2F	Input Output Control	Override actuator/sensor states.
0x31	Routine Control	0x01 Start Routine; 0x02 Stop Routine; 0x03 Request Results.
0x34	Request Download	Start software download.
0x35	Request Upload	Retrieve memory block contents.
0x36	Transfer Data	Send software/memory block.
0x37	Request Transfer Exit	Conclude download/upload session.
0x38	Request File Transfer	Optional file transfer extension.
0x3E	Tester Present	Keep diagnostic session alive.
0x83	Access Timing Parameters	Adjust response timing.
0x84	Secured Data Transmission	Cryptographically protected exchange.
0x85	Control DTC Setting	Enable/disable fault code recording.
0x86	Response on Event	Trigger responses based on ECU-defined events.
0x87	Link Control	Adjust link characteristics (e.g., baud rate).