INTERNATIONAL STANDARD

ISO 14229-1

Second edition
2013-03-15

Road vehicles — Unified diagnostic services (UDS) —

Part 1: **Specification and requirements**

Véhicules routiers — Services de diagnostic unifiés (SDU) —

Partie 1: Spécification et exigences

Reference number
ISO 14229-1:2013(E)

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 14229-1 was prepared by Technical Committee ISO/TC 22, *Road vehicles*, Subcommittee SC 3, *Electrical and electronic equipment*.

This second edition cancels and replaces the first edition (ISO 14229-1:2006), which has been technically revised.

ISO 14229 consists of the following parts, under the general title *Road vehicles — Unified diagnostic services (UDS)*:

— *Part 1: Specification and requirements*

— *Part 2: Session layer services*

— *Part 3: Unified diagnostic services on CAN implementation (UDSonCAN)*

— *Part 4: Unified diagnostic services on FlexRay implementation (UDSonFR)*

— *Part 5: Unified diagnostic services on Internet Protocol implementation (UDSonIP)*

— *Part 6: Unified diagnostic services on K-Line implementation (UDSonK-Line)*

The following part is under preparation:

— *Part 7: Unified diagnostic services on Local Interconnect Network implementation (UDSonLIN)*

The titles of future parts will be drafted as follows:

— *Part n: Unified diagnostic services on … implementation (UDSon…)*

Introduction

ISO 14229 has been established in order to define common requirements for diagnostic systems, whatever the serial data link is.

To achieve this, ISO 14229 is based on the Open Systems Interconnection (OSI) Basic Reference Model in accordance with ISO 7498-1 and ISO/IEC 10731, which structures communication systems into seven layers. When mapped on this model, the services used by a diagnostic tester (client) and an Electronic Control Unit (ECU, server) are broken into the following layers in accordance with Table 1:

— Application layer (layer 7), unified diagnostic services specified in ISO 14229-1, ISO 14229-3 UDSonCAN, ISO 14229-4 UDSonFR, ISO 14229-5 UDSonIP, ISO 14229-6 UDSonK-Line, ISO 14229-7 UDSonLIN, further standards and ISO 27145-3 WWH-OBD.

— Presentation layer (layer 6), vehicle manufacturer specific, ISO°27145-2 WWH-OBD.

— Session layer services (layer 5) specified in ISO 14229-2.

— Transport layer services (layer 4), specified in ISO 15765-2 DoCAN, ISO 10681-2 Communication on FlexRay, ISO 13400-2 DoIP, ISO 17987-2 LIN, ISO 27145-4 WWH-OBD.

— Network layer services (layer 3), specified in ISO 15765-2 DoCAN, ISO 10681-2 Communication on FlexRay, ISO 13400-2 DoIP, ISO 17987-2 LIN, ISO 27145-4 WWH-OBD.

— Data link layer (layer 2), specified in ISO 11898-1, ISO 11898-2, ISO 17458-2, ISO 13400-3, IEEE 802.3, ISO 14230-2, ISO 17987-3 LIN and further standards, ISO 27145-4 WWH-OBD.

— Physical layer (layer 1), specified in ISO 11898-1, ISO 11898-2, ISO 17458-4, ISO 13400-3, IEEE 802.3, ISO 14230-1, ISO 17987-4 LIN and further standards, ISO 27145-4 WWH-OBD.

NOTE The diagnostic services in this standard are implemented in various applications e.g. Road vehicles – Tachograph systems, Road vehicles – Interchange of digital information on electrical connections between towing and towed vehicles, Road vehicles – Diagnostic systems, etc. It is required that future modifications to this standard provide long-term backward compatibility with the implementation standards as described above.

Table 1 — Example of diagnostic/programming specifications applicable to the OSI layers

Applicability	OSI seven layer	Enhanced diagnostics services						WWH-OBD
Seven layer according to ISO/IEC 7498-1 and ISO/IEC 10731	Application (layer 7)	ISO 14229-1, ISO 14229-3 UDSonCAN, ISO 14229-4 UDSonFR, ISO 14229-5 UDSonIP, ISO 14229-6 UDSonK-Line, ISO 14229-7 UDSonLIN, further standards						ISO 27145-3
	Presentation (layer 6)	vehicle manufacturer specific						ISO 27145-2
	Session (layer 5)	ISO 14229-2						
	Transport (layer 4)	ISO 15765-2	ISO 10681-2	ISO 13400-2	Not applicable	ISO 17987-2	further standards	ISO 27145-4
	Network (layer 3)						further standards	
	Data link (layer 2)	ISO 11898-1, ISO 11898-2	ISO 17458-2	ISO 13400-3, IEEE 802.3	ISO 14230-2	ISO 17987-3	further standards	
	Physical (layer 1)		ISO 17458-4		ISO 14230-1	ISO 17987-4	further standards	

Road vehicles — Unified diagnostic services (UDS) —

Part 1: Specifications and requirements

1 Scope

This part of ISO 14229 specifies data link independent requirements of diagnostic services, which allow a diagnostic tester (client) to control diagnostic functions in an on-vehicle Electronic Control Unit (ECU, server) such as an electronic fuel injection, automatic gear box, anti-lock braking system, etc. connected to a serial data link embedded in a road vehicle.

It specifies generic services, which allow the diagnostic tester (client) to stop or to resume non-diagnostic message transmission on the data link.

This part of ISO 14229 does not apply to non-diagnostic message transmission on the vehicle's communication data link between two Electronic Control Units. However, this part of ISO 14229 does not restrict an in-vehicle on-board tester (client) implementation in an ECU in order to utilize the diagnostic services on the vehicle's communication data link to perform bidirectional diagnostic data exchange.

This part of ISO 14229 does not specify any implementation requirements.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 14229-2, *Road vehicles — Unified diagnostic services (UDS) — Part 2: Session layer services*

3 Terms, definitions, symbols and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1.1
boot manager
part of the boot software that executes immediately after an ECU power on or reset whose primary purpose is to check whether a valid application is available to execute as compared to transferring control to the reprogramming software

NOTE The boot manager may also take into account other conditions for transitioning control to the reprogramming software.

3.1.2
boot memory partition
area of the server memory in which the boot software is located

3.1.3
boot software
software which is executed in a special part of server memory which is used primarily to boot the ECU and perform server programming

NOTE 1 This area of memory is not erased during a normal programming sequence and must execute when the server application is missing or otherwise deemed invalid to always ensure the capability to reprogram the server.

NOTE 2 See also 3.1.1 and 3.1.17.

3.1.4
client
function that is part of the tester and that makes use of the diagnostic services

NOTE A tester normally makes use of other functions such as data base management, specific interpretation, human-machine interface.

3.1.5
diagnostic data
data that is located in the memory of an electronic control unit which may be inspected and/or possibly modified by the tester

NOTE 1 Diagnostic data includes analogue inputs and outputs, digital inputs and outputs, intermediate values and various status information.

NOTE 2 Examples of diagnostic data are vehicle speed, throttle angle, mirror position, system status, etc. Three types of values are defined for diagnostic data:

— the current value: the value currently used by (or resulting from) the normal operation of the electronic control unit;

— a stored value: an internal copy of the current value made at specific moments (e.g. when a malfunction occurs or periodically); this copy is made under the control of the electronic control unit;

— a static value: e.g. VIN.

The server is not obliged to keep internal copies of its data for diagnostic purposes, in which case the tester may only request the current value.

NOTE 3 Defining a repair shop or development testing session selects different server functionality (e.g. access to all memory locations may only be allowed in the development testing session).

3.1.6
diagnostic routine
routine that is embedded in an electronic control unit and that may be started by a server upon a request from the client

NOTE It could either run instead of a normal operating program, or could be enabled in this mode and executed with the normal operating program. In the first case, normal operation for the server is not possible. In the second case, multiple diagnostic routines may be enabled that run while all other parts of the electronic control unit are functioning normally.

3.1.7
diagnostic service
information exchange initiated by a client in order to require diagnostic information from a server or/and to modify its behaviour for diagnostic purpose

3.1.8
diagnostic session
state within the server in which a specific set of diagnostic services and functionality is enabled

3.1.9
diagnostic trouble code
DTC
numerical common identifier for a fault condition identified by the on-board diagnostic system

3.1.10
ECU
electronic control unit, containing at least one server

NOTE Systems considered as Electronic Control Units include Anti-lock Braking System (ABS) and Engine Management System.

3.1.11
functional unit
set of functionally close or complementary diagnostic services

3.1.12
integer type
simple type with distinguished values which are the positive and the negative whole numbers, including zero

NOTE The range of type integer is not specified within this part of ISO 14229.

3.1.13
local client
client that is connected to the same local network as the server and is part of the same address space as the server

3.1.14
local server
server that is connected to the same local network as the client and is part of the same address space as the client

3.1.15
OSI
open systems interconnection

3.1.16
permanent DTC
diagnostic trouble code (DTC) that remains in non-volatile memory, even after a clear DTC request, until other criteria (typically regulatory) are met (e.g. the appropriate monitors for each DTC have successfully passed)

NOTE Refer to the relevant legislation for all necessary requirements.

3.1.17
record
one or more diagnostic data elements that are referred to together by a single means of identification

NOTE A snapshot including various input/output data and trouble codes is an example of a record.

3.1.18
remote server
server that is not directly connected to the main diagnostic network

NOTE 1 A remote server is identified by means of a remote address. Remote addresses represent an own address space that is independent from the addresses on the main network.

NOTE 2 A remote server is reached via a local server on the main network. Each local server on the main network can act as a gate to one independent set of remote servers. A pair of addresses must therefore always identify a remote server: one local address that identifies the gate to the remote network and one remote address identifying the remote server itself.

3.1.19
remote client
client that is not directly connected to the main diagnostic network

NOTE 1 A remote client is identified by means of a remote address.

NOTE 2 Remote addresses represent an own address space that is independent from the addresses on the main network.

3.1.20
reprogramming software
part of the boot software that allows for reprogramming of the electronic control unit

3.1.21
security
mechanism for protecting vehicle modules from "unauthorized" intrusion through a vehicle diagnostic data link

3.1.22
server
function that is part of an electronic control unit and that provides the diagnostic services

NOTE This international standard differentiates between the server (i.e. the function) and the electronic control unit so that this standard remains independent from the implementation.

3.1.23
supported DTC
diagnostic trouble code which is currently configured/calibrated and enabled to execute under pre-defined vehicle conditions

3.1.24
tester
system that controls functions such as test, inspection, monitoring, or diagnosis of an on-vehicle electronic control unit and may be dedicated to a specific type of operator (e.g. an off-board scan tool dedicated to garage mechanics, an off-board test tool dedicated to assembly plants, or an on-board tester)

NOTE The tester is also referenced as the client.

3.2 Abbreviated terms

.con	service primitive .confirmation
.ind	service primitive .indication
.req	service primitive .request
A_PCI	application layer protocol control information
ECU	electronic control unit
EDR	event data recorder
N/A	not applicable
NR_SI	negative response service identifier
NRC	negative response code
OSI	open systems interconnection

RA	remote address
SA	source address
SI	service identifier
TA	target address
TA_type	target address type

4 Conventions

This part of ISO 14229 is based on the conventions discussed in the OSI Service Conventions (ISO/IEC 10731:1994) as they apply for diagnostic services.

These conventions specify the interactions between the service user and the service provider. Information is passed between the service user and the service provider by service primitives, which may convey parameters.

The distinction between service and protocol is summarised in Figure 1.

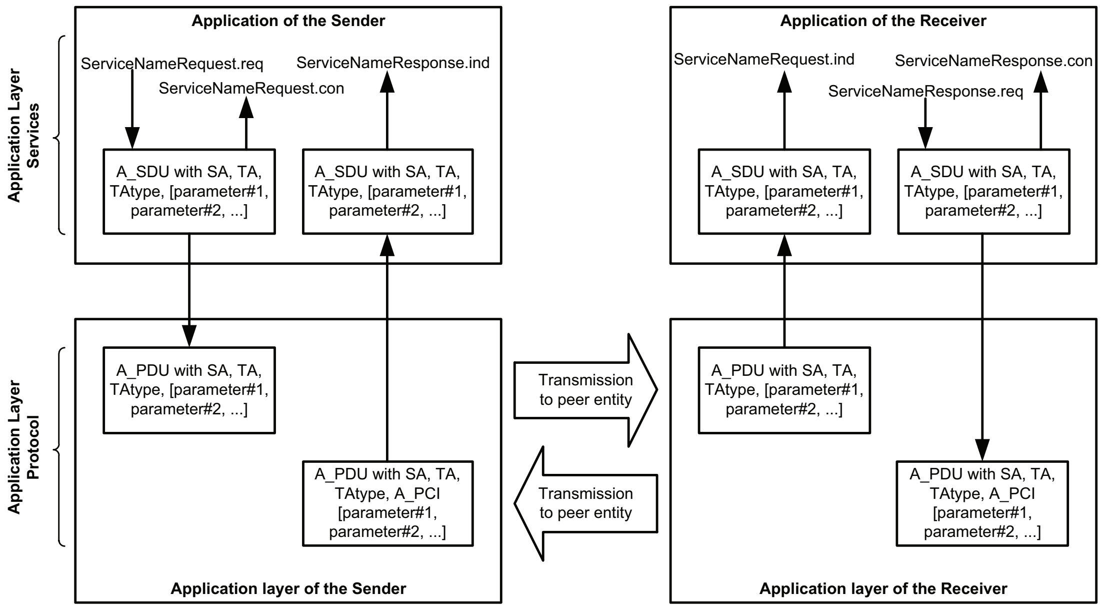
Application Layer Services
Application Layer Protocol
Application of the Sender
ServiceNameRequest.req
ServiceNameResponse.ind
ServiceNameRequest.con
A_SDU with SA, TA, TAtype, [parameter#1, parameter#2, ...]
A_SDU with SA, TA, TAtype, [parameter#1, parameter#2, ...]
A_PDU with SA, TA, TAtype, [parameter#1, parameter#2, ...]
A_PDU with SA, TA, TAtype, A_PCI [parameter#1, parameter#2, ...]
Application layer of the Sender
Transmission to peer entity
Transmission to peer entity
Application of the Receiver
ServiceNameRequest.ind
ServiceNameResponse.con
ServiceNameResponse.req
A_SDU with SA, TA, TAtype, [parameter#1, parameter#2, ...]
A_SDU with SA, TA, TAtype, [parameter#1, parameter#2, ...]
A_PDU with SA, TA, TAtype, [parameter#1, parameter#2, ...]
A_PDU with SA, TA, TAtype, A_PCI [parameter#1, parameter#2, ...]
Application layer of the Receiver

Figure 1 — The services and the protocol

This part of ISO 14229 defines both confirmed and unconfirmed services.

The confirmed services use the six service primitives request, req_confirm, indication, response, rsp_confirm and confirmation.

The unconfirmed services use only the request, req_confirm and indication service primitives.

For all services defined in this part of ISO 14229 the request and indication service primitives always have the same format and parameters. Consequently for all services the response and confirmation service primitives (except req_confirm and rsp_confirm) always have the same format and parameters. When the service primitives are defined in this International Standard, only the request and response service primitives are listed.

5 Document overview

Figure 2 depicts the implementation of UDS document reference according to OSI model.

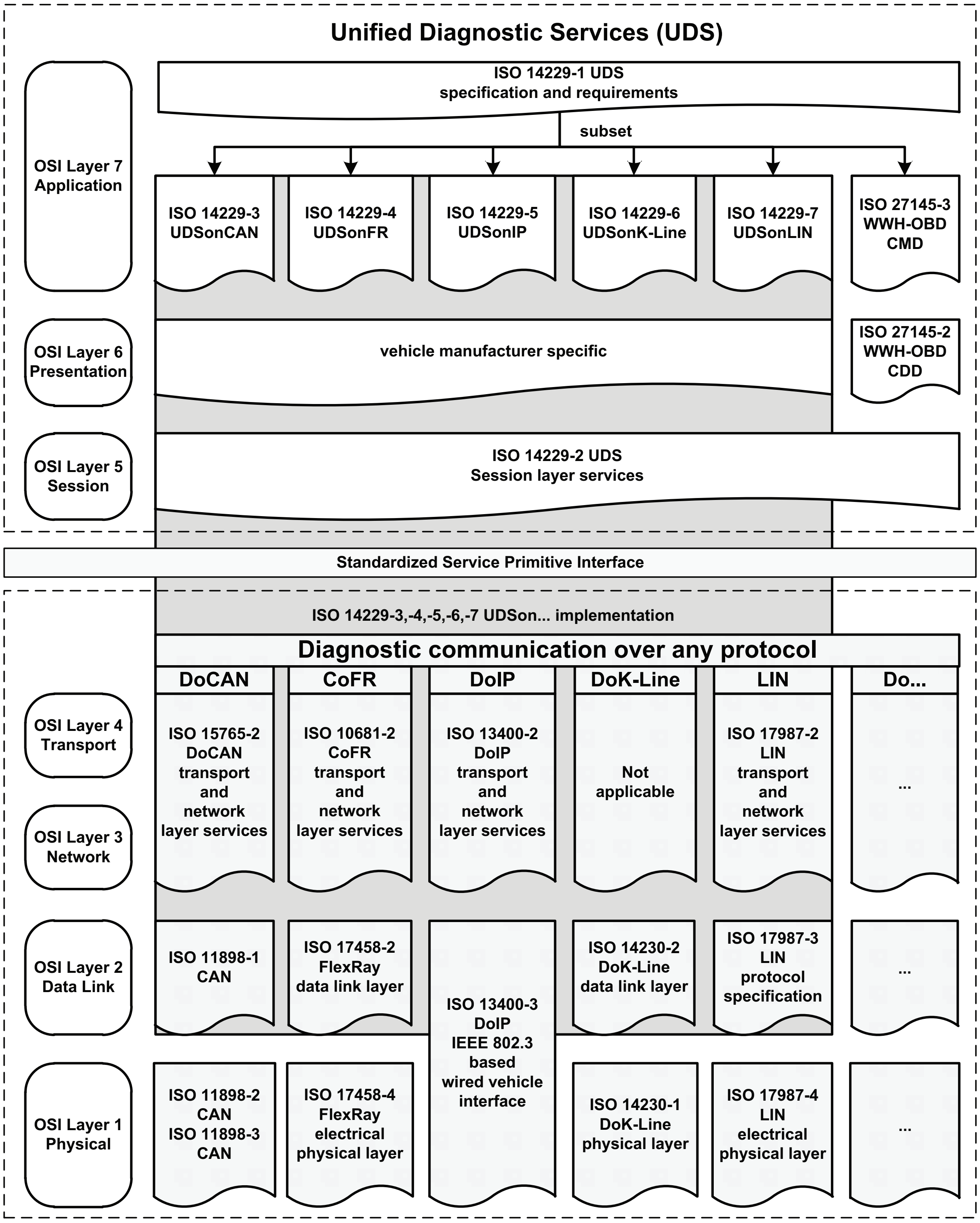
Unified Diagnostic Services (UDS)
OSI Layer 7 Application
ISO 14229-1 UDS specification and requirements
subset
ISO 14229-3 UDSonCAN
ISO 14229-4 UDSonFR
ISO 14229-5 UDSonIP
ISO 14229-6 UDSonK-Line
ISO 14229-7 UDSonLIN
ISO 27145-3 WWH-OBD CMD
OSI Layer 6 Presentation
vehicle manufacturer specific
ISO 27145-2 WWH-OBD CDD
OSI Layer 5 Session
ISO 14229-2 UDS Session layer services
Standardized Service Primitive Interface
ISO 14229-3,-4,-5,-6,-7 UDSon... implementation
Diagnostic communication over any protocol
DoCAN
CoFR
DoIP
DoK-Line
LIN
Do...
OSI Layer 4 Transport
OSI Layer 3 Network
ISO 15765-2 DoCAN transport and network layer services
ISO 10681-2 CoFR transport and network layer services
ISO 13400-2 DoIP transport and network layer services
Not applicable
ISO 17987-2 LIN transport and network layer services
...
OSI Layer 2 Data Link
ISO 11898-1 CAN
ISO 17458-2 FlexRay data link layer
ISO 13400-3 DoIP IEEE 802.3 based wired vehicle interface
ISO 14230-2 DoK-Line data link layer
ISO 17987-3 LIN protocol specification
...
OSI Layer 1 Physical
ISO 11898-2 CAN ISO 11898-3 CAN
ISO 17458-4 FlexRay electrical physical layer
ISO 14230-1 DoK-Line physical layer
ISO 17987-4 LIN electrical physical layer
...

Figure 2 — Implementation of UDS document reference according to OSI model

6 Application layer services

6.1 General

Application layer services are usually referred to as diagnostic services. The application layer services are used in client-server based systems to perform functions such as test, inspection, monitoring or diagnosis of on-board vehicle servers. The client, usually referred to as external test equipment, uses the application layer services to request diagnostic functions to be performed in one or more servers. The server, usually a function that is part of an ECU, uses the application layer services to send response data, provided by the requested diagnostic service, back to the client. The client is usually an off-board tester, but can in some systems also be an on-board tester. The usage of application layer services is independent from the client being an off-board or on-board tester. It is possible to have more than one client in the same vehicle system.

The service access point of the diagnostics application layer provides a number of services that all have the same general structure. For each service, six service primitives are specified:a **service request primitive**, used by the client function in the diagnostic tester application, to pass data about a requested diagnostic service to the diagnostics application layer;

— a **service request primitive**, used by the client function in the diagnostic tester application, to pass data about a requested diagnostic service to the diagnostics application layer;

— a **service request-confirmation primitive**, used by the client function in the diagnostic tester application, to indicate that the data passed in the service request primitive is successfully sent on the vehicle communication bus the diagnostic tester is connected to

— a **service indication primitive**, used by the diagnostics application layer, to pass data to the server function of the ECU diagnostic application;

— a **service response primitive**, used by the server function in the ECU diagnostic application, to pass response data provided by the requested diagnostic service to the diagnostics application layer;

— a **service response-confirmation primitive**, used by the server function in the ECU diagnostic application, to indicate that the data passed in the service response primitive is successfully sent on the vehicle communication bus the ECU received the diagnostic request on;

— a **service confirmation primitive** used by the diagnostics application layer to pass data to the client function in the diagnostic tester application.

Figure 3 depicts the Application layer service primitives - confirmed service.

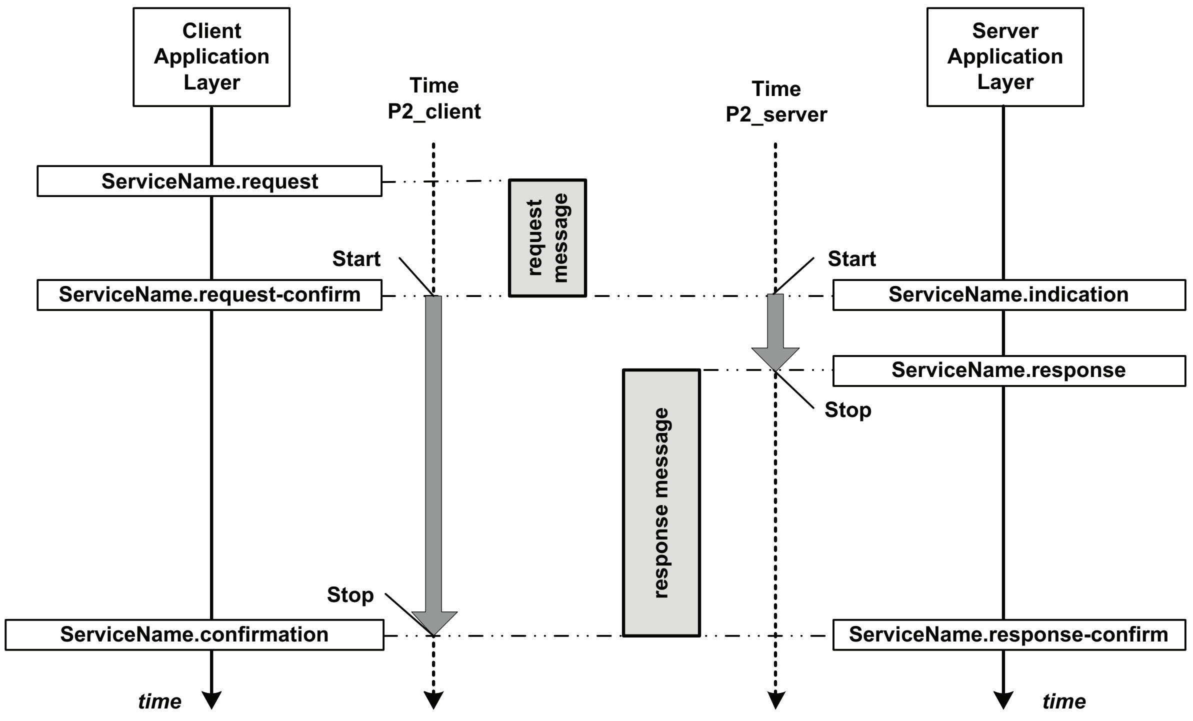

Figure 3 — Application layer service primitives - confirmed service

Figure 4 depicts the application layer service primitives - unconfirmed service.

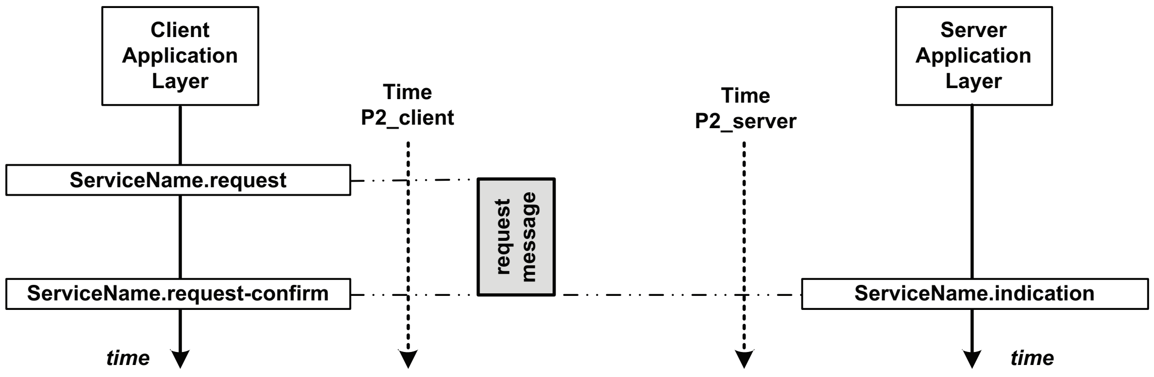

Figure 4 — Application layer service primitives - unconfirmed service

For a given service, the request-confirmation primitive and the response-confirmation primitive always have the same service data unit. The purpose of these service primitives is to indicate the completion of an earlier request or response service primitive invocation. The service descriptions in this International Standard will not make use of those service primitives, but the data link specific implementation documents might use those to define e.g. service execution reference points (e.g. the ECUReset service would invoke the reset when the response is completely transmitted to the client which is indicated in the server via the service response-confirm primitive).

6.2 Format description of application layer services

Application layer services can have two different formats depending on how the vehicle diagnostic system is configured. The format of the application layer service is controlled by parameter A_Mtype.

If the vehicle system is configured so that the client can address all servers by using the A_SA and A_TA address parameters, the default format of application layer services shall be used. This implies A_Mtype = diagnostics.

If the vehicle system is configured so that the client needs address information in addition to the A_SA and A_TA address parameters allowing to address certain servers, the remote format of application layers services shall be used. This implies A_Mtype = remote diagnostics.

The different formats for application layer services are specified in 6.3.

6.3 Format description of service primitives

6.3.1 General definition

All application layer services have the same general format. Service primitives are written in the form:

```
service_name.type  (
                   parameter A, parameter B, parameter C
                   [,parameter 1, ...]
                   )
```

Where:

— "service_name" is the name of the diagnostic service (e.g. DiagnosticSessionControl),

— "type" indicates the type of the service primitive (e.g. request),

— "parameter A, ..." is the A_SDU (Application layer Service Data Unit) as a list of values passed by the service primitive (addressing information),

— "parameter A, parameter B, parameter C" are mandatory parameters that shall be included in all service calls,

— "[,parameter 1, ...]" are parameters that depend on the specific service (e.g. parameter 1 can be the diagnosticSession for the DiagnosticSessionControl service). The brackets indicate that this part of the parameter list may be empty.

6.3.2 Service request and service indication primitives

For each application layer service, service request and service indication primitives are specified according to the following general format:

```
service_name.request   (
                      A_MType,
                      A_SA,
                      A_TA,
                      A_TA_type,
                      [A_AE],
                      A_Length,
                      A_Data[,parameter 1, ...],
                      )
```

The request primitive is used by the client function in the diagnostic tester application, to initiate the service and pass data about the requested diagnostic service to the application layer.

```
service_name.indication   (
                      A_MType,
                      A_SA,
                      A_TA,
                      A_TA_type,
                      [A_AE],
                      A_Length
                      A_Data[,parameter 1, ...],
                      )
```

The indication primitive is used by the application layer, to indicate an internal event which is significant to the ECU diagnostic application and pass data about the requested diagnostic service to the server function of the ECU diagnostic application.

The request and indication primitive of a specific application layer service always have the same parameters and parameter values. This means that the values of individual parameters shall not be changed by the communicating peer protocol entities of the application layer when the data is transmitted from the client to the server. The same values that are passed by the client function in the client application to the application layer in the service request call shall be received by the server function of the diagnostic application from the service indication of the peer application layer.

6.3.3 Service response and service confirm primitives

For each confirmed application layer service, service response and service confirm primitives are specified according to the following general format:

```
service_name.response  (
                        A_Mtype,
                        A_SA,
                        A_TA,
                        A_TA_type,
                        [A_AE],
                        A_Length
                        A_Data[,parameter 1, ...],
                        )
```

The response primitive is used by the server function in the ECU diagnostic application, to initiate the service and pass response data provided by the requested diagnostic service to the application layer.

```
service_name.confirm   (
                        A_Mtype,
                        A_SA,
                        A_TA,
                        A_TA_type,
                        [A_AE],
                        A_Length
                        A_Data[,parameter 1, ...],
                        )
```

The confirm primitive is used by the application layer to indicate an internal event which is significant to the client application and pass results of an associated previous service request to the client function in the diagnostic tester application. It does not necessarily indicate any activity at the remote peer interface, e.g if the requested service is not supported by the server or if the communication is broken.

The response and confirm primitive of a specific application layer service always have the same parameters and parameter values. This means that the values of individual parameters shall not be changed by the communicating peer protocol entities of the application layer when the data is transmitted from the server to the client. The same values that are passed by the server function of the ECU diagnostic application to the application layer in the service response call shall be received by the client function in the diagnostic tester application from the service confirmation of the peer application layer.

For each response and confirm primitive two different service data units (two sets of parameters) will be specified.

— A positive response and positive confirm primitive shall be used with the first service data unit if the requested diagnostic service could be successfully performed by the server function in the ECU.

— A negative response and confirm primitive shall be used with the second service data unit if the requested diagnostic service failed or could not be completed in time by the server function in the ECU.