



US 20170024943A1

(19) **United States**(12) **Patent Application Publication**  
**Wodecki et al.**(10) **Pub. No.: US 2017/0024943 A1**(43) **Pub. Date: Jan. 26, 2017**(54) **SYSTEM AND METHOD FOR SERVICE  
ASSESSMENT****Related U.S. Application Data**(63) Continuation-in-part of application No. 14/219,755,  
filed on Mar. 19, 2014, now Pat. No. 9,514,580.(71) Applicant: **Cummins, Inc.**, Columbus, IN (US)**Publication Classification**(72) Inventors: **Arthur R. Wodecki**, Columbus, IN  
(US); **Russell Martin Poling, JR.**,  
Columbus, IN (US); **Ronald R.**  
**Chapman**, Columbus, IN (US); **Setu**  
**Madhavi Namburu**, Columbus, IN  
(US); **Michael A. Wodecki**, White Bear  
Lake, MN (US); **Brent A. Engel**, Hope,  
IN (US); **Helena L. Weaks**, Maineville,  
OH (US); **Steven R. Collins**,  
Columbus, IN (US); **David E. Schisler**,  
Columbus, IN (US)(51) **Int. Cl.**  
**G07C 5/08** (2006.01)  
**G05B 23/02** (2006.01)  
**G07C 5/00** (2006.01)(52) **U.S. Cl.**  
CPC ..... **G07C 5/0808** (2013.01); **G07C 5/006**  
(2013.01); **G07C 5/008** (2013.01); **G07C**  
**5/0825** (2013.01); **G05B 23/0294** (2013.01)(73) Assignee: **Cummins, Inc.**, Columbus, IN (US)(21) Appl. No.: **15/282,755**(22) Filed: **Sep. 30, 2016**(57) **ABSTRACT**Systems, apparatuses, and methods disclosed provide for  
service assessment. The method includes receiving at least  
one fault code of an engine control unit (ECU); determining  
at least one root cause based on the at least one fault code  
of the ECU; determining an estimated labor time for fixing  
the at least one root cause; and outputting the at least one  
root cause and the estimated labor time to a display.

		10,000	1,000	100	10	1	0.1	0.01	
		1	2	3	4	5	6	7	
		Voltage = 1, or Not = 0	ECU = 1, or Not = 0	Component = 7, System = 3, or Effect = 0	Sensor supply = 9; Circuitry COR = 7; Smart = 3; Other = 1	1 "point" for each interaction	Engine = 1, or Aftertreatment = 0	Fault Code / 1,000	
Fault Code	Subsystem	Voltage	ECU	Component/ System	Failure Type	Interactions	Upstream/ Downstream	Fault Code/ 1,000	Priority
1673	DEF Tank	0	0	7	1	4	0	1.673	714.01673

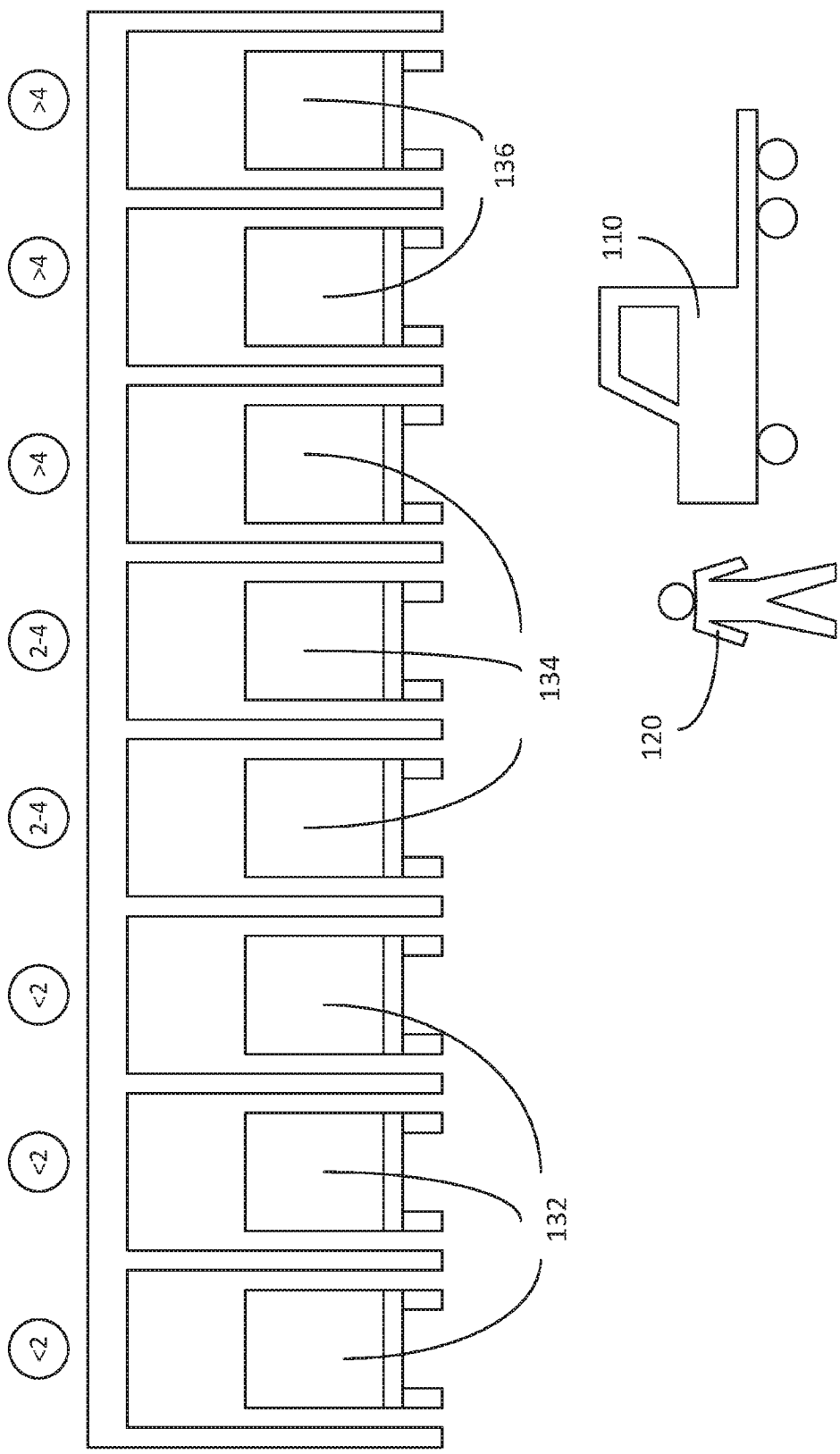


FIG. 1(A)

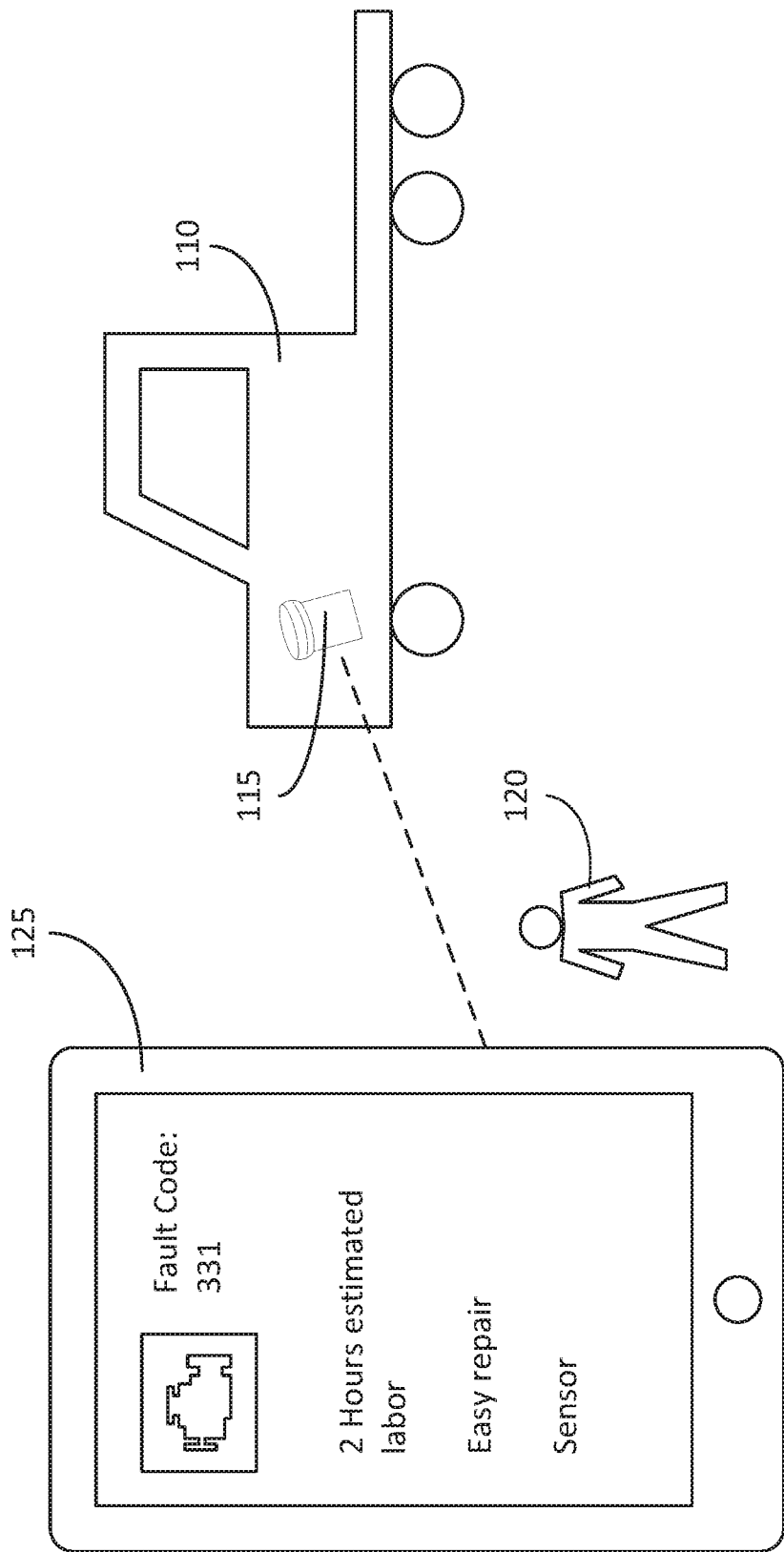


FIG. 1(B)

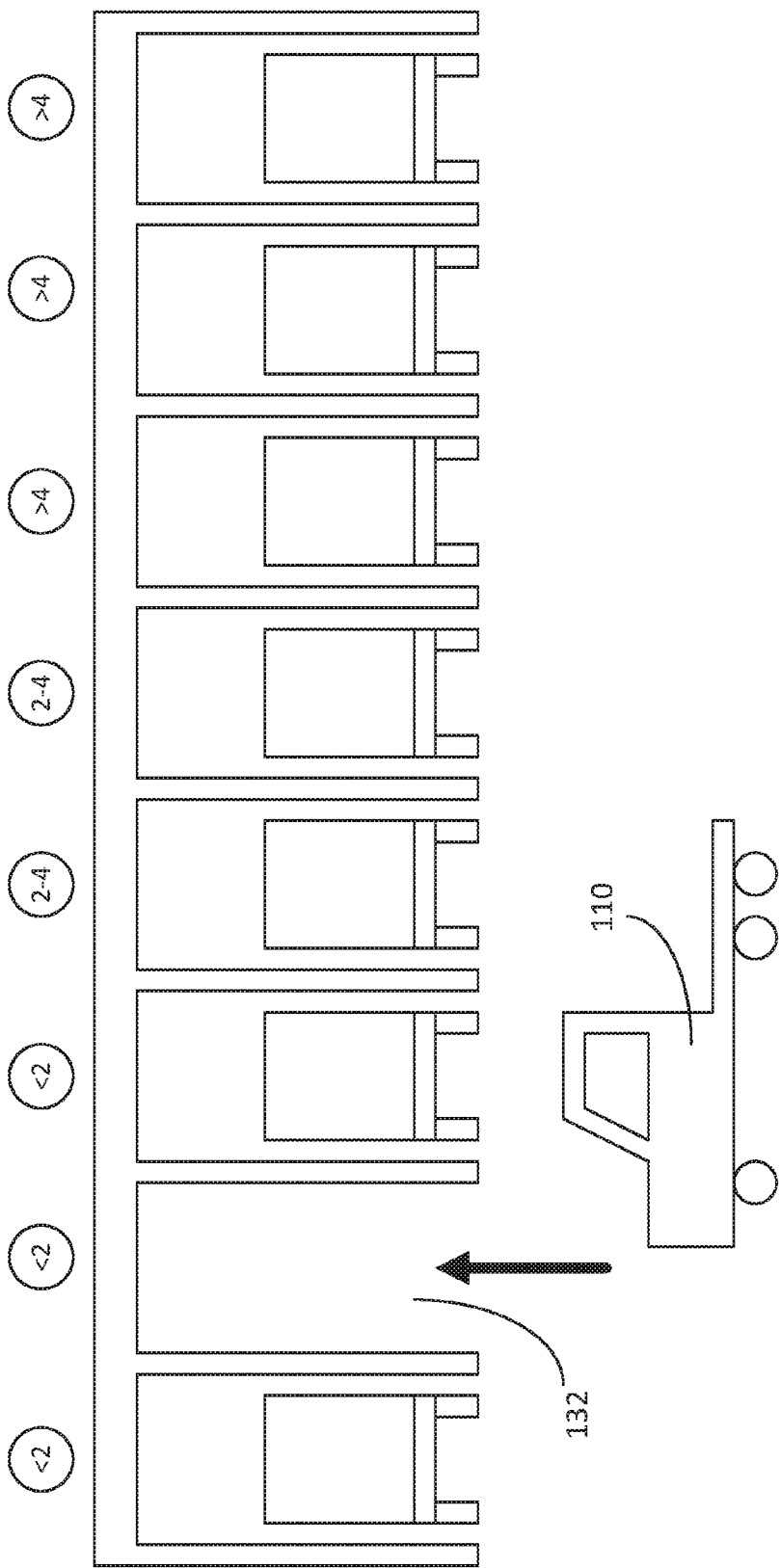


FIG. 1(C)

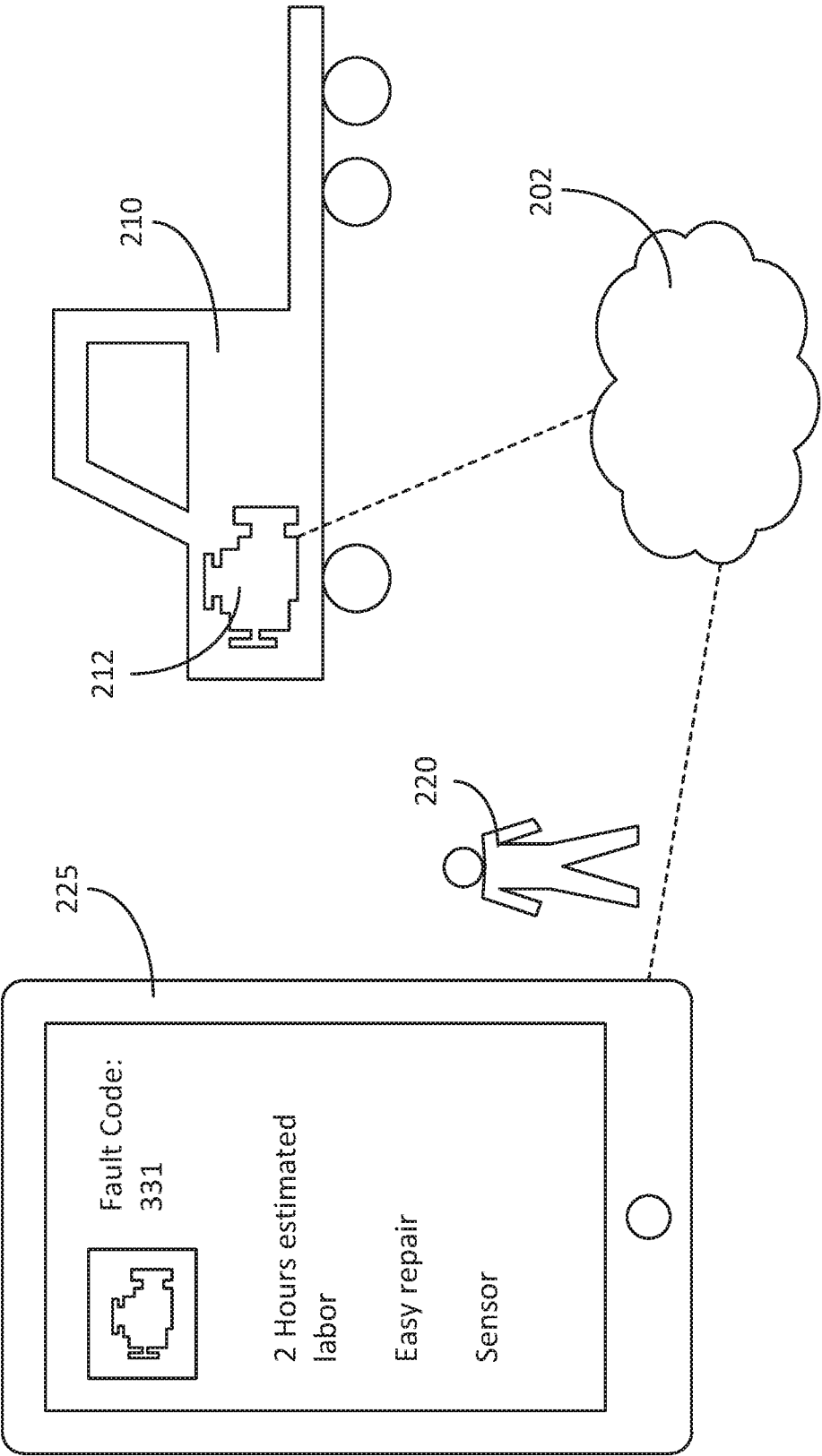
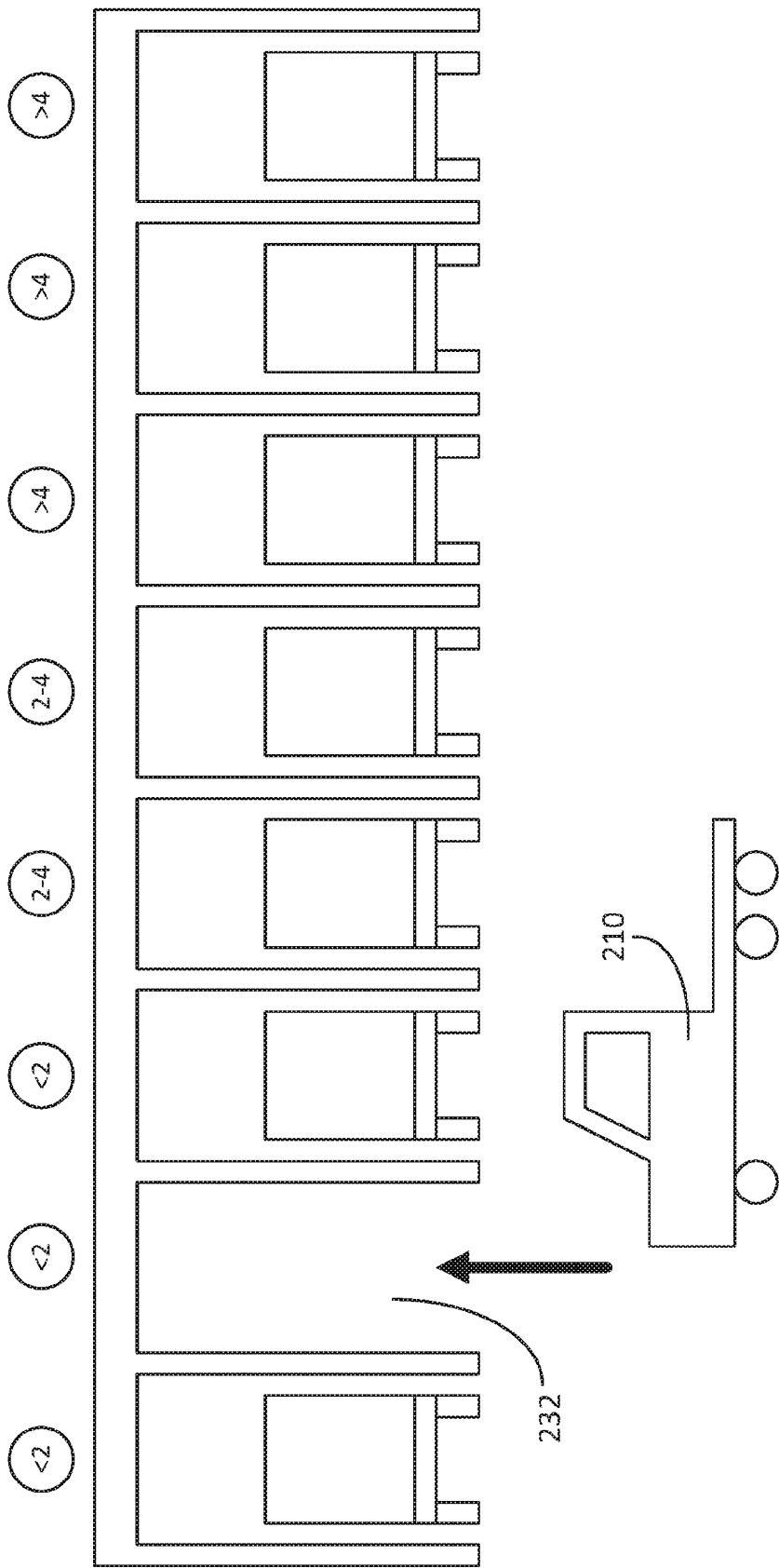


FIG. 2(A)



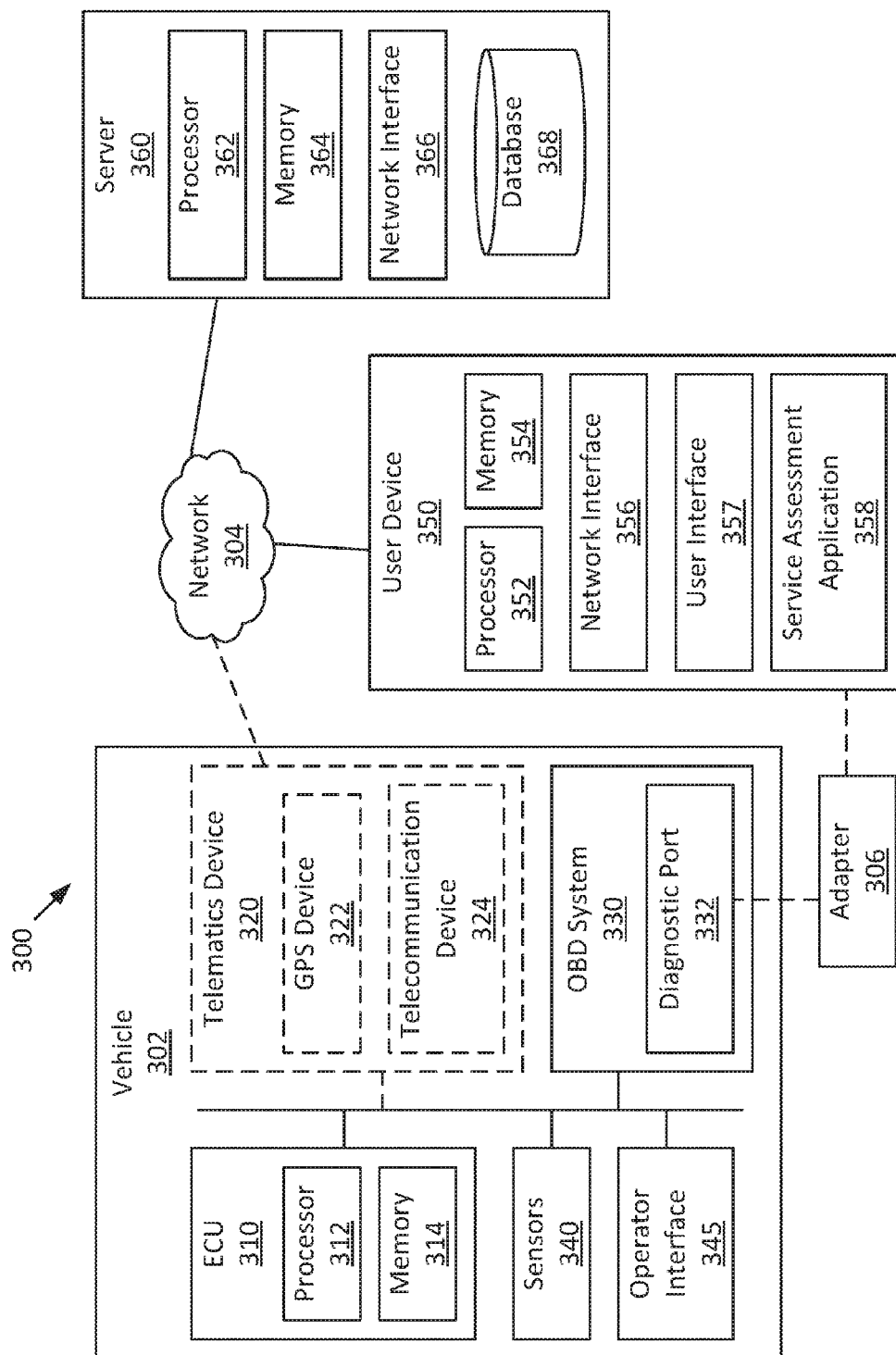


FIG. 3

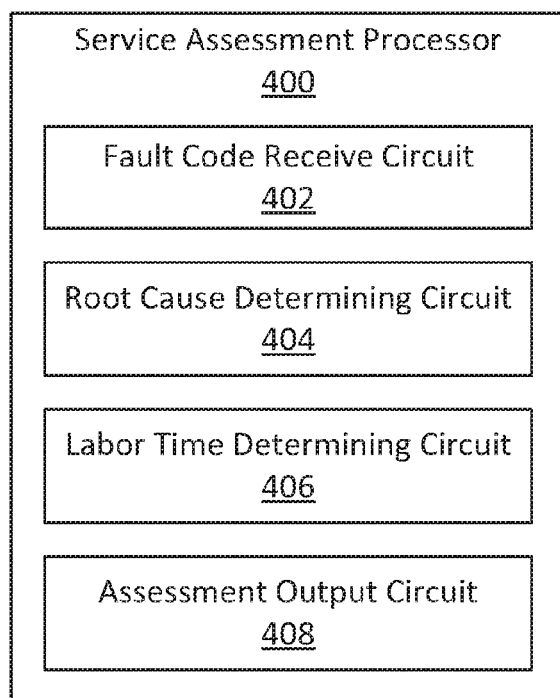


FIG. 4

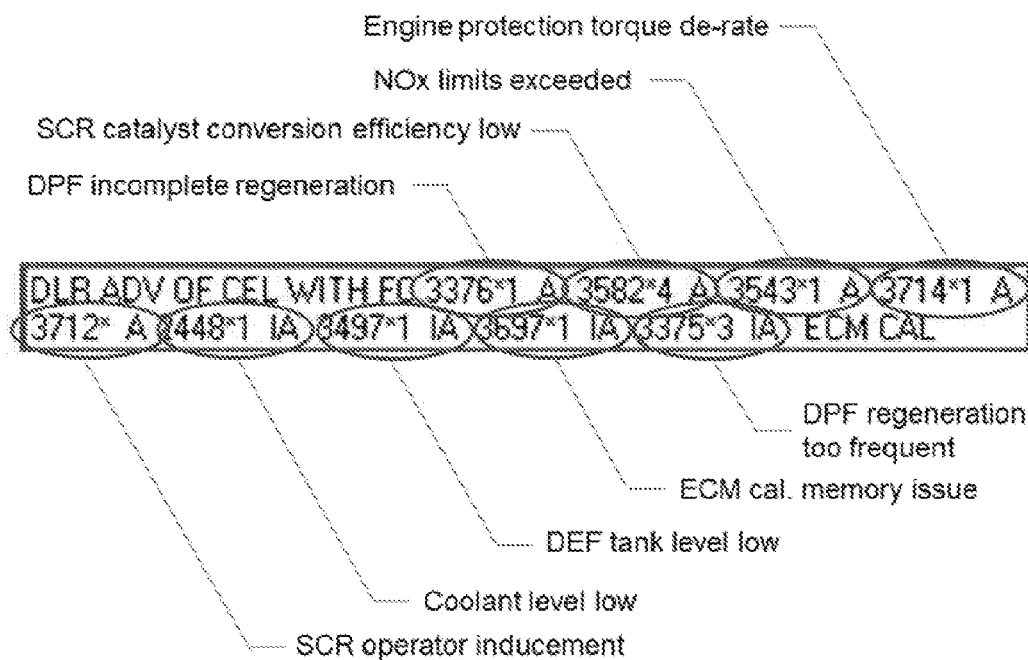


FIG. 5



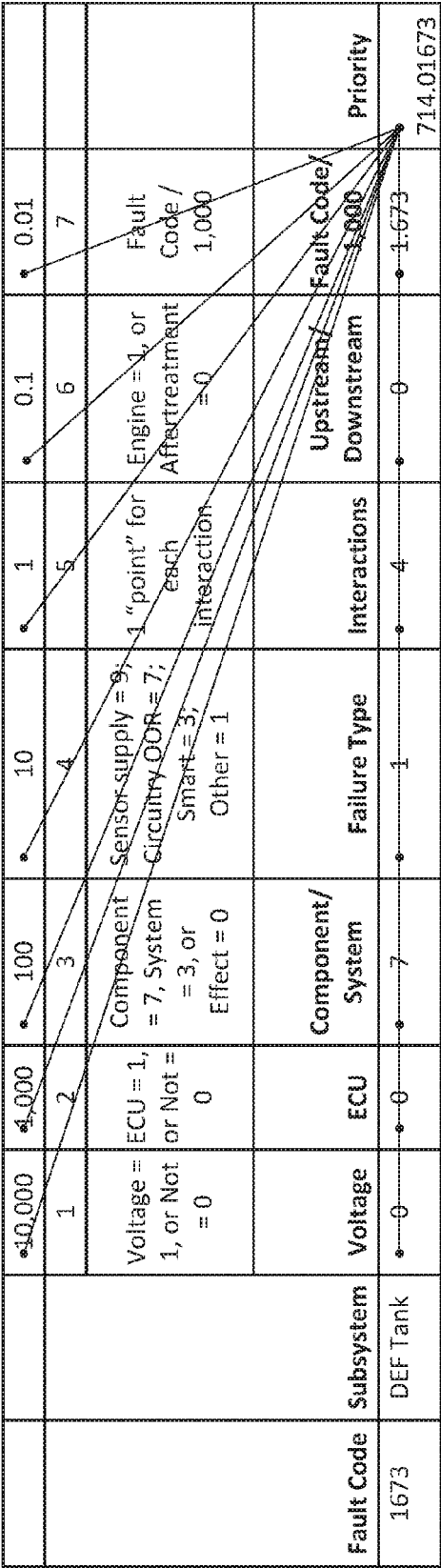


FIG. 6

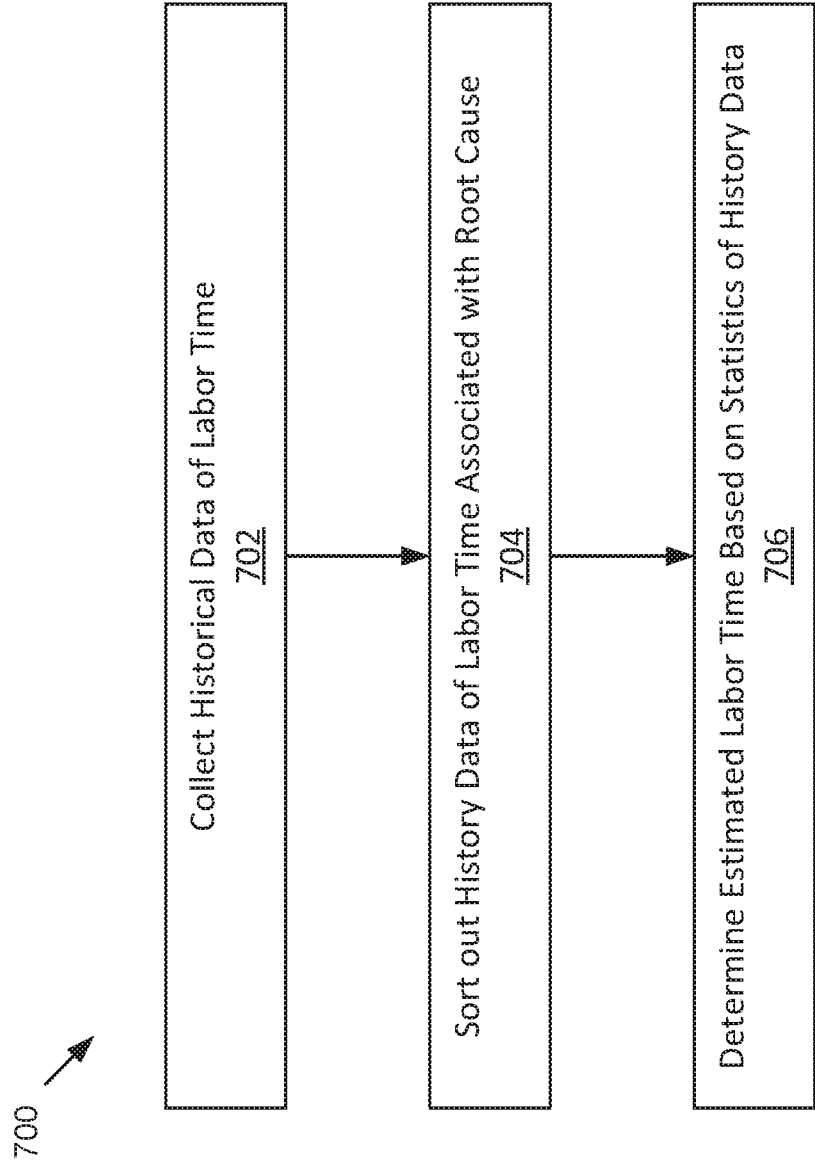


FIG. 7

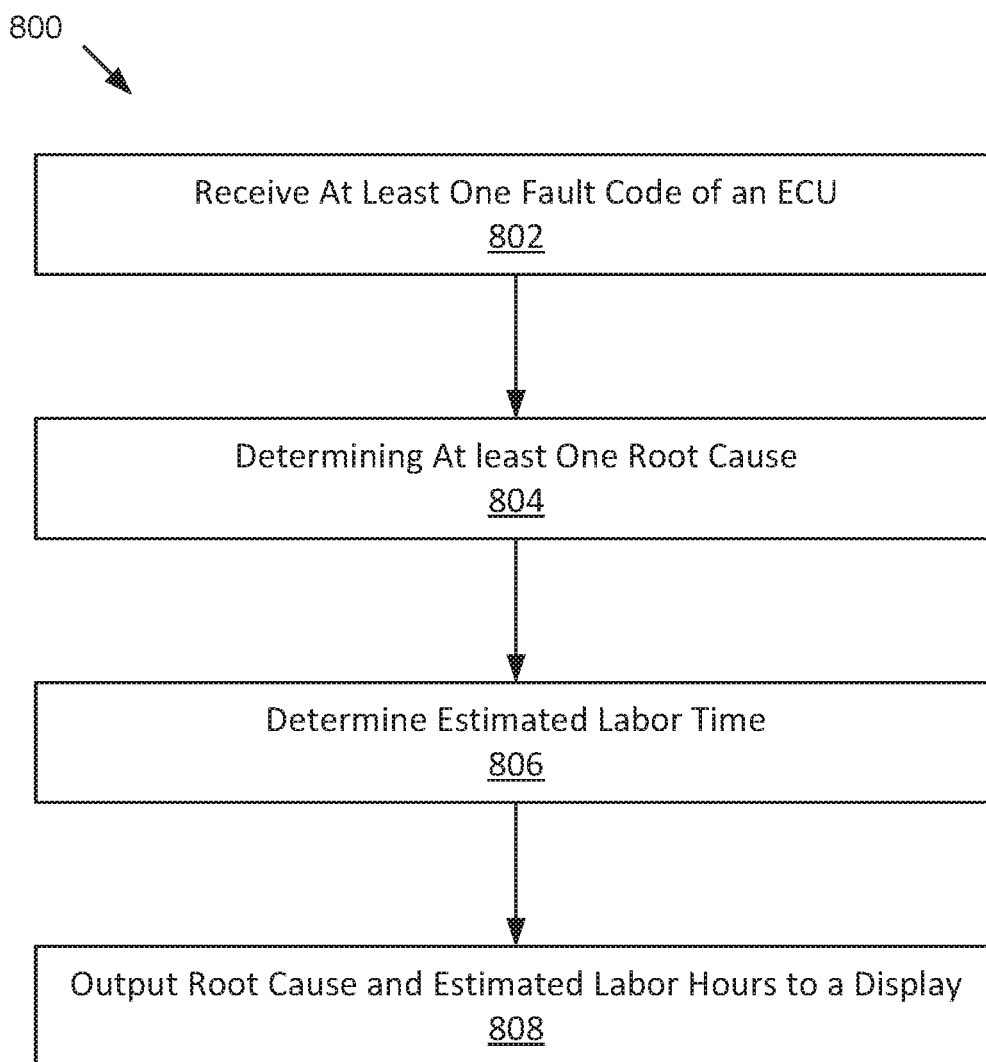


FIG. 8

## SYSTEM AND METHOD FOR SERVICE ASSESSMENT

### CROSS-REFERENCE TO RELATED APPLICATIONS

**[0001]** This application is a continuation-in-part of U.S. patent application Ser. No. 14/219,755, entitled "FAULT CODE HIERARCHY SYSTEM," filed Mar. 19, 2014, which is incorporated herein by reference in its entirety.

### TECHNICAL FIELD

**[0002]** The present disclosure relates to systems and methods for assessing service events associated with equipment.

### BACKGROUND

**[0003]** A customer may take equipment (e.g., a vehicle) to a service provider location (e.g., a customer service site, an OEM service site, a dealership, a distributorship, or a repair shop) when some abnormal symptoms of the equipment are observed. For example, the check engine light is on, the engine generates abnormal noise or smoke, the engine is hard to start, the engine runs rough, to name a few. At the service provider location, a technician may provide a preliminary assessment regarding possible cause of the fault. Based on the assessment, the vehicle may be directed to a repair bay for further diagnosis and repair. In some occasions, the vehicle might sit a few days at the service provider location, yet the repair might take only a few hours. Improvements on the efficiency of the service assessment are generally desired.

### SUMMARY

**[0004]** One embodiment relates to a system for service assessment. The system comprises an input circuit structured to receive at least one fault code of an engine control unit (ECU); a root cause determination circuit structured to determine at least one root cause based on the at least one fault code of the ECU; a labor time estimation circuit structured to determine an estimated labor time for fixing the at least one root cause; and an output circuit structured to output the at least one root cause and the estimated labor time to a display.

**[0005]** Another embodiment relates to a method for service assessment. The method comprises receiving at least one fault code of an engine control unit (ECU); determining at least one root cause based on the at least one fault code of the ECU; determining an estimated labor time for fixing the at least one root cause; and outputting the at least one root cause and the estimated labor time to a display.

**[0006]** Yet another embodiment relates to a user device. The user device comprises an input port structured to receive at least one fault code of an engine control unit (ECU), a controller coupled to the input port, and a display coupled to the controller. The controller is structured to determine at least one root cause based on the at least one fault code, and determine an estimated labor time for fixing the at least one root cause. The display is structured to display the at least one root cause and the estimated labor time.

**[0007]** These and other features, together with the organization and manner of operation thereof, will become apparent from the following detailed description when taken in conjunction with the accompanying drawings.

### BRIEF DESCRIPTION OF THE FIGURES

**[0008]** FIGS. 1(A)-1(C) are schematic diagrams of a scenario of providing service assessment for a vehicle at a service provider location, according to an example embodiment.

**[0009]** FIGS. 2(A)-2(B) are schematic diagrams of a scenario of providing service assessment for a vehicle at a service provider location, according to another example embodiment.

**[0010]** FIG. 3 is a block diagram of a computing environment for assessing service event associated with a vehicle according to an example embodiment.

**[0011]** FIG. 4 is a block diagram of a processor for assessing service event associated with a vehicle, according to an example embodiment.

**[0012]** FIG. 5 is a diagram of fault codes read from a vehicle, according to an example embodiment.

**[0013]** FIG. 6 is a diagram of priority number determination for a fault code number 1673, according to an example embodiment.

**[0014]** FIG. 7 is a flow diagram of a method for determining estimated labor time, according to an example embodiment.

**[0015]** FIG. 8 is a flow diagram of a method for service assessment, according to an example embodiment.

### DETAILED DESCRIPTION

**[0016]** For the purposes of promoting an understanding of the principles of the disclosure, reference will now be made to the embodiments illustrated in the drawings and specific language will be used to describe the same. It will nevertheless be understood that no limitation of the scope of the disclosure is thereby intended, any alterations and further modifications in the illustrated embodiments, and any further applications of the principles of the disclosure as illustrated therein as would normally occur to one skilled in the art to which the disclosure relates are contemplated herein.

**[0017]** Referring to the Figures generally, the various embodiments disclosed herein relate to systems and methods for assessing service events associated with equipment. A customer may take the equipment (e.g., a vehicle) to a service provider location (e.g., a customer service site, an OEM service site, a dealership, a distributorship, or a repair shop) when some abnormal symptoms (e.g., check engine light on, abnormal noise or smoke, hard start, rough operation) of the equipment are observed. Many service providers commit to their customers that a preliminary assessment be performed within certain amount of time (e.g., two hours) from when the customer arrives at the service provider location. Using the systems and methods disclosed herein, the preliminary assessment may be performed more quickly. In particular, historical data of diagnosis and repair on the type of equipment is collected and built into a database over time. When faults occur to the equipment, the database can be queried to find the diagnosis and repair data relating to equipment of similar abnormal symptoms. Based on the historical data, one can determine what part(s) might have caused the faults and how hard it would be to repair the equipment. In the case for vehicles, a software application is provided on a user device (e.g., smartphone, tablet). The application can retrieve fault information (e.g., fault codes) for the vehicle, determine possible cause of the fault, deter-

mine the estimated labor time for diagnosing and repairing the vehicle, and display the possible cause and the estimated labor time. Appropriate service event can be scheduled for the vehicle based on the displayed information.

[0018] For example, FIGS. 1(A)-1(C) show a scenario of assessing service event for a vehicle 110 at a service provider location, according to an example embodiment. When the vehicle 110 arrives, a service advisor 120, who does not have to be a service technician, greets the customer. The service advisor uses a user mobile device (e.g., smartphone) 125, installed with a software application for service assessment, to retrieve fault code(s) from an on-board diagnostic (OBD) system 115 on the vehicle 110 through, for example, an adapter. The application determines that a sensor might have caused the fault which might be easy to fix, and that the estimated labor time is two hours. The information is displayed on the user device 125. There are a plurality of bays 132, 134, and 136 on the service provider location assigned to service events that take various lengths of time. For example, bays 132 are assigned to service events that might take no longer than two hours, bays 134 are assigned to service events that might take two to four hours, and bays 136 are assigned to service events that might take longer than four hours. Based on the assessment that the service might take two hours, the vehicle 110 is directed to a bay 132 that is currently available.

[0019] FIGS. 2(A)-2(B) show a scenario of providing service assessment for a vehicle 210 at a service provider location, according to another example embodiment. As shown, the vehicle 210 has a telematics device 212, which can transmit the fault code information to the user device 225 via a network 202. Thus the user device 225 can receive the fault code information and determine possible cause of the fault and estimated labor hours before the vehicle 210 arrives at the service provider location. When the vehicle 210 arrives, the service advisor 220 directs the vehicle 210 to a corresponding bay 232 based on the reading of the possible source and estimated labor hours from the user device 225.

[0020] Referring now to FIG. 3, a block diagram of a computing environment 300 for assessing service event associated with a vehicle is shown according to an example embodiment. The computing environment 300 includes a vehicle 302, a user device 350, and a server 360. In some embodiments, the vehicle 302, the user device 350, and the server 360 can communicate with each other via a network 304. In some embodiments, the vehicle 302 can be coupled to the user device 350 through an adapter 306 at a service provider location, for example, a customer service site, an OEM service site, a dealer site, a distributor site, or a repair shop.

[0021] The vehicle 302 may be any type of passenger or commercial vehicle, such as a car, truck, sport utility vehicle, cross-over vehicle, van, minivan, automobile, tractor-trailer, and so on. Moreover, the vehicle 302 may include other types of vehicles such as a motorcycle, plane, helicopter, locomotive, or railway car. Although the methods and systems are illustrated with reference with a vehicle, it shall be understood that the methods and systems disclosed herein can apply to other equipment, such as power generator sets, appliances, etc. The vehicle 302 is shown to include an engine control unit (ECU) 310, a telematics device 320, an on-board diagnostic (OBD) system 330, sensors 340, and an operator interface 345. Components of the vehicle 302

may communicate with each other via any number of wired or wireless connections. For example, a wired connection may include a serial cable, a fiber optic cable, a CATS cable, or any other form of wired connection. In comparison, a wireless connection may include the Bluetooth, Wi-Fi, cellular, radio, etc. In one embodiment, components of the vehicle 302 are connected to a vehicle network such as a control area network (CAN) or a manufacturer proprietary network.

[0022] The ECU 310 includes a processor 312 and a memory 314. The memory 314 stores various instructions that, when executed by the processor 312, control the operation of various components and/or subsystems of the vehicle 102. For example, the ECU 310 may include an electronic fuel injection control unit, engine mobilizer control unit, aftertreatment system control unit, etc. The processor 312 may be implemented as a general-purpose processor, an application specific integrated circuit (ASIC), one or more field programmable gate arrays (FPGAs), a digital signal processor (DSP), a group of processing components, or other suitable electronic processing components. The memory 314 may include one or more tangible, non-transient volatile memory or non-volatile memory, e.g., NVRAM, RAM, ROM, Flash Memory, hard disk storage, etc.). Moreover, the memory 314 may include database components, object code components, script components, or any other type of information structure.

[0023] The OBD system 330 may be structured to diagnose the performance of the components and subsystems of the vehicle 302, such as ECU, ABS system, heating/air conditioning system, brake system, transmission system, and so on. Sensors 340 are positioned throughout the vehicle 302 and monitor the operational status and condition of a wide range of components and subsystems of the vehicle 302. For example, sensors 340 can measure engine temperature/speed/load, battery voltage, aftertreatment system emission, tire pressure, fuel tank level, etc. The OBD system 330 may receive data indicative of operational status and conditions of the components and subsystems monitored by the sensors 340. Based on the data, the OBD system 330 diagnoses malfunctions or deterioration in performance of the components and subsystems. In one example, if a monitored parameter falls outside of a normal range of operation, the OBD system 330 may generate a fault code indicative of the abnormal parameter. For instance, if the engine coolant level falls outside of a predefined normal operating range, the OBD system 330 will issue a fault code indicating that the engine coolant level is low. In another example, if a subsystem or component fails or is unable to send its output due to an open or short-circuit, the OBD system 330 may generate a fault code indicative of the failure to read the parameter. For instance, wires may be shorted high or shorted low, which would generate wiring faults; some part of the wiring harness might be disconnected with the key on, which would generate datalink-related faults. In further embodiments, the fault code generated is stored in the memory 314. It shall be noted that different manufactures may have different fault code systems.

[0024] The OBD system 330 may indicate the diagnosed fault at the operator interface 345 of the vehicle 302. The operator interface 345 enables an operator of the vehicle 302 to read status and conditions of the vehicle 302. For example, the operator interface 345 may include, but is not

limited, an interactive display (e.g., a touchscreen, etc.), a dashboard, a control panel, etc. The operator interface **345** may display the fault information reported by the OBD system **330** for the operator to read. For example, the check engine light on the dashboard may be turned on when the OBD system **330** diagnoses a fault of the engine. The tire air pressure light may be turned on when the OBD system **330** determines the tire air pressure is outside of the normal operational range, and so on.

**[0025]** The OBD system **330** includes a diagnostic port **332** through which an off-board service tool can access the OBD system **330**. The off-board service tool may be used at a service provider location, which may be an OEM handheld scan tool/code reader or a computing device (e.g., smartphone, laptop) installed with the off-board diagnostic software (e.g., Cummins Insite®). The diagnostic port **332** is a hardware interface, such as a 16-pin serial port, a D-shaped serial port disposed, for example, underneath the dashboard of the vehicle **302**. In some embodiments in which the vehicle **302** is a truck, the diagnostic port **332** may be a 6 or 9-pin serial port disposed in the truck's interior. Communication between the OBD system **330** and the off-board diagnostic service tool may follow different protocols, for example but not limited to, SAE J1939, J1708/J1587, J1850 VPW, J1850 VPWM, ISO 9141-2, Keyword 2000, and CAN. Besides the fault information (e.g., fault codes), other vehicle conditions and status information such as vehicle identification number (VIN), mileage, engine speed, etc. can also be obtained via the diagnostic port **332**. The VIN is unique to each vehicle and includes information about its make, model, year (and month) of production, and serial number.

**[0026]** In some embodiments, a telematics device **320** is installed on the vehicle **302**. A telematics system is the convergence of telecommunications and information processing, which involves sending, receiving, and processing information via telecommunication devices. The telematics device **320** is structured to transmit information relating to the vehicle **302** (e.g., the fault information) obtained onboard to a remote server **360** and/or the user device **350** via the network **304**. In some embodiments, the server **360** can immediately diagnose the fault and provide actionable information (e.g., fault alert) to the user instantly. In some embodiments, the telematics device **320** is an OEM device embedded in the vehicle **302**. In other embodiments, the telematics device **320** is an aftermarket standalone device, e.g., a telematics box coupled to the vehicle **302** through, for example, the diagnostic port **332**. The telematics device **120** may obtain the raw data acquired by the sensors **340** and/or the fault information determined by the OBD system **330**. In some embodiments, the telematics device **320** may integrate certain telecommunication functions. For example, a navigation system within the vehicle **302** may be included in the telematics device **320**.

**[0027]** The telematics device **320** includes a GPS device **322** and a telecommunication device **324**. The GPS device **322** tracks the location of the vehicle **302** (e.g., latitude and longitude data, elevation data, etc.). The telecommunication device **324** communicates with external devices over the network **304**. Although not shown in FIG. 3, the telecommunication device **324** may include an antenna, a radio frequency (RF) transceiver, and a subscriber identity module (SIM). The telecommunication device **324** may follow any type of mobile communications protocol, for example but

not limited to, cellular, satellite, radio, Wi-Fi, WiMax, Bluetooth, Zigbee, GSM, GPRS, LTE, and the like.

**[0028]** The server **360** may receive vehicle-related data from the telematics device **320**, store and analyze the data, and notify the customer of the faults through email, text message, etc. For example, at the moment an engine system fault occurs, the telematics device **320** may instantly transmit key engine system and GPS data to the server **360**. The server **360** may immediately analyze the data and provide actionable information to the operator, e.g., provide the fault information and even a location of a service site at the operator interface **345** and/or through email or text message. In further embodiments, besides telematics devices on vehicles, the server **360** receives vehicle-related information from other sources. For example, the server **360** can receive information relating to warranty claims, and/or repair information from service providers around the world via the network **304**. In some embodiments, the server **360** is implemented as a central computing system hosted by an engine manufacture, a vehicle manufacturer, a telematics provider, an OEM, or multiple parties. In some embodiments, the telematics server **360** is implemented as a cloud network including multiple computing systems, which can share and transfer vehicle information and data store, and coordinate to process the received data.

**[0029]** The server **360** includes a processor **362**, a memory **364**, a network interface **366**, and a database **368**. The memory **364** stores various instructions that, when executed by the processor **362**, control the operation of the server **360**. The network interface **366** allows the server **360** to send and receive data to and from external devices via the network **304**. The network interface **366** may be a wireless network interface that communicates with a wireless communication protocol (e.g., 802.11a/b/g/n, Bluetooth®, ZigBee®, CDMA, GSM, LTE, WiMax, etc.) or a wired communication protocol (e.g., Ethernet, USB, Thunderbolt®, etc.). The database **368** is structured to receive and store, hold, and otherwise serve as a repository for fault information, repair information, and other vehicle-related data received from information sources. In some embodiments, the database **368** may be a separate component relative to the telematics server **360**. For example, due to the potential high volume or quantity of data stored by the database **368**, the database **368** may be formed or constructed from two or more server-based storage components stored over two or more remote geographic locations. In some embodiments, the database **368** may also include one or more classification and/or categorization functions (e.g., logic processing, etc.). The classification function may sort, categorize, or otherwise classify each piece of received data from vehicles according to, for example, the vehicle identification number (VIN), the make/model of vehicles, or any other suitable parameters. In further embodiments, the database **368** may store product recalls, historical problem history, and related known current problems of the vehicle **302**.

**[0030]** A service assessment application **358** is installed on the user device **350** for determining possible cause and estimated labor hours for diagnosing and repairing the vehicle **302** based on fault information of the vehicle **302**. In some embodiments, the user device **350** is coupled to the diagnostic port **332** of the vehicle through the adapter **306** and reads the fault information (e.g., fault codes) from the diagnostic port **332**. In some embodiments where the vehicle **302** has the telematics device **320**, the user device **350**

receives the fault information transmitted by the telematics device 320. The user device 350 may be smartphone, a tablet, a personal digital assistant (PDA), a laptop computer, and so on. The user device 350 can be used by a service advisor at a service provider location. The user device 350 includes a processor 352, memory 354, and a network interface 133. Memory 354 stores various program instructions (e.g., service assessment instructions) that, when executed by the processor 352, control the operation of the user device 350. The network interface 356 allows the user device 350 to send and receive data to and from external devices via the network 304. The network interface 356 may be a wireless network interface that communicates with a wireless communication protocol (e.g., 802.11a/b/g/n, Bluetooth®, ZigBee®, CDMA, GSM, LTE, WiMax, etc.) or a wired communication protocol (e.g., Ethernet, USB, Thunderbolt®, etc.). The user interface 357 may include a display and an input. In some embodiments, the display and input are integrated in a touchscreen display. The user device 350 is installed with the service assessment application 358, which is a software executable by the processor 352 to implement at least some or all of the functions described herein.

[0031] The adapter 306 provides an interface between the OBD system 330 and the user device 350. The adapter 306 may be plugged into the diagnostic port 332 to read the fault information, and transmit the fault information to the user device 350 through a wired connection (e.g., USB) or a wireless connection (e.g., Bluetooth®).

[0032] The network 304 may include private networks, public networks, or a combination thereof. In some embodiments, the network 304 includes the Internet. The network 304 may be a combination of wireless and wired networks. The wireless network may be any type of wireless network, for example, a satellite or cellular network using protocols such as Global System for Mobile Communications (GSM), Code Division Multiple Access (CDMA), General Packet Radio Service (GPRS), Long Term Evolution (LTE), High Rate Packet Data (HRPD), Wi-Fi, Personal Communications Service (PCS), etc. The wired network may be any type of wired network, for example, Ethernet network, Local Talk, Fiber Distributed Data Interface (FDDI), etc.

[0033] Referring now to FIG. 4, a schematic diagram of a service assessment processor 400 for assessing service events associated with vehicles is shown. The service assessment processor 400 can be implemented on the user device 350, the server 360, or a combination thereof. In other words, the functions of the service assessment processor 400 as described herein can be executed by the processor 352 of the user device 350, the processor 362 of the server 360, or a combination thereof. The service assessment processor 400 includes various circuits for completing the activities described herein, each circuit performing a particular functionality. The electronic circuitry on the service assessment processor 400 may be considered to be shared across each circuit. It shall be understood that the service assessment processor 400 may include any number of circuits for completing the functions and activities described herein. For example, the activities of multiple circuits may be combined as a single circuit, as an additional circuit(s) with additional functionality, etc. Further, it should be understood that the service assessment processor 400 may further control other activity beyond the scope of the present disclosure.

[0034] Certain operations of the service assessment processor 400 described herein include operations to interpret and/or to determine one or more parameters. Interpreting or determining, as utilized herein, includes receiving values by any method known in the art, including at least receiving values from a datalink or network communication, receiving an electronic signal (e.g. a voltage, frequency, current, or PWM signal) indicative of the value, receiving a computer generated parameter indicative of the value, reading the value from a memory location on a non-transient computer readable storage medium, receiving the value as a run-time parameter by any means known in the art, and/or by receiving a value by which the interpreted parameter can be calculated, and/or by referencing a default value that is interpreted to be the parameter value.

[0035] The service assessment processor 400 includes a fault code receive circuit 402, a root cause determining circuit 404, a labor time determining circuit 406, and an assessment output circuit 408. Through the circuits 402-408, the service assessment processor 400 is structured to receive fault code(s) from the vehicle 302, determine at least one possible root cause based on the fault code(s), determine an estimated labor time for fixing the root cause, and output the possible root cause and estimated labor time to a display (e.g., the user interface 357 of the user device 350).

[0036] The fault code receive circuit 402 is structured to receive fault code(s) from the vehicle 302. In some embodiments, the fault code(s) are received from the diagnostic port 332 through the adapter 306. In some embodiments, the fault code(s) are received from the telematics device 320 on the vehicle 302. FIG. 5 shows fault codes read from a vehicle according to an example embodiment. Each fault code represents an error that has occurred in the vehicle operation. For example, fault code "3376" represents "DPF incomplete regeneration," "3582" represents "SCR catalyst conversion efficiency low," "3543" represents "NOx limits exceeded," and so on. In some embodiments, the fault code receive circuit 402 receives other vehicle status and conditions information together with the fault code(s), such as VIN, make, model, year (and month) of production, serial number, mileage, engine speed, etc.

[0037] The root cause determining circuit 404 is structured to determine at least one root cause based on the fault code(s) received. In some embodiments, the root cause is represented by one fault code from multiple fault codes received from the vehicle 302. In some embodiment, the root cause is represented by the name of a component, the failure of which might have caused the received fault code(s). In some embodiments, the root cause determining circuit 404 analyzes the received fault code on its own to determine the root cause. In some embodiments, the root cause determining circuit 404 sends the fault code(s) to the server 360 via the network 304. The server 360 analyzes the fault codes and sends the determined root cause back to the root cause determining circuit 404 on the user device 350.

[0038] In some embodiments, the root cause determining circuit 404 includes a fault code analyzer as disclosed in the U.S. patent application Ser. No. 14/219,755, which is incorporated herein in its entirety by reference. The fault code analyzer is used where an engine has generated multiple fault codes. The fault code analyzer can put the multiple fault codes in sequence and point to up to three root cause fault codes. In some embodiments, the fault code analyzer is included in the root cause determining circuit 404 on the

user device 350. In some embodiments, the fault code analyzer is implemented on the server 360.

[0039] In particular, when multiple fault codes are received, the fault code analyzer categorizes what the fault code relates to (e.g., voltage, ECM, etc.) and determines a priority for each fault code. Typically, the priority ranking is based on a priority number. The higher the priority number, the more likely that fault code is the root cause of the problem. The fault code with the highest priority number may be designated as the first root cause. The fault code analyzer may list any and all interaction fault codes for the first root cause fault code. As used herein, an interaction fault code refers to a fault code caused by the same issue that has caused the root cause fault code. Determining that a fault code is an interaction fault code may be based on many characteristics (e.g., each fault code is related to the same engine or vehicle component). If any fault codes remain, the fault code analyzer will designate a second root cause fault code, based on the next highest priority number for the unaccounted fault codes and provide the remaining interaction fault codes for the second root cause. In some embodiments, the fault code analyzer analyzes inactive fault codes in addition to active fault codes, which increases the possibility of identifying the correct and all root causes. As used herein, active fault codes refer to fault codes that indicate a current fault condition; inactive fault codes indicate a condition that has occurred since the last time fault data was cleared, but is not presently active. By accounting for inactive fault codes, a relatively greater chance of correctly diagnosing the engine malfunction exists.

[0040] According to one embodiment, the fault code analyzer determines the priority number for any fault code based on the fault code hierarchy categories below:

- [0041] 1. Voltage ( $a_1=1$  or 0,  $w_1=10,000$ )(Yes—1, No—0)
- [0042] 2. ECU ( $a_2=1$  or 0,  $w_2=1,000$ )(Yes—1, No—0)
- [0043] 3. Fault code type ( $a_3=7, 3$ , or 0,  $w_3=100$ )(Component—7, System—3, or Effect—0)
- [0044] 4. Failure type ( $a_4=9, 7, 3$ , or 1,  $w_4=10$ )(Sensor Supply—9, Circuitry Out-of-Range—7, Smart Devices—3, Other—1)
- [0045] 5. Number of interactions ( $a_5$ =Number of Interactions,  $w_5=1$ )
- [0046] 6. Location ( $a_6=1$  or 0,  $w_6=0.1$ )(Engine—1, Aftertreatment—0)
- [0047] 7. Fault code number ( $a_7$ =(Fault Code Number)/1000,  $w_7=0.01$ )

[0048] The hierarchy above provides the basis for a formula, algorithm, and method of generating the priority numbers. According to one configuration, the priority number determination is as follows, where the “a” variables correspond with the category values and the “w” variables correspond with the weighting (“multiplier”) for each category:

$$\text{Priority No.} = a_1 * w_1 + a_2 * w_2 + a_3 * w_3 + a_4 * w_4 + a_5 * w_5 + a_6 * w_6 + a_7 * w_7$$

[0049] As explained more fully herein, the above category ranking is configured to identify the most likely root cause fault code(s), such that the likely root cause fault code(s) can be addressed first by a technician. According to one embodiment, the most likely root cause fault code corresponds with the highest value priority number. Thus, the weighting values shown above indicate which category most largely impacts the priority number value (e.g., fault codes that

correspond with voltage related issues are more likely to be determined by the fault code analyzer to be root cause fault codes than fault codes that are unrelated to voltage conditions because of the relatively large  $w_1$  value). According to various other embodiments, the weighting used (i.e., “w” values) may be adjusted to reflect, for example, various desires of the technician or user. For example, if the technician is concerned with engine-related fault codes, the technician may weight (increase the multiplier) of category 6 (location) higher in order to determine a likely root cause for engine-related fault codes. Accordingly, the seven hierarchical categories provide the basis for determining the root cause fault code in order to troubleshoot the root malfunction/condition first. As such, technicians may not need to search for the root cause malfunction and may be efficiently able to service likely root problems of the engine or vehicle, which saves time and money. The example above is only an example embodiment. Accordingly, fewer, different, and/or additional categories may be used for each priority number determination and the weighting may be different than previously described.

[0050] The priority number determination is more fully described as follows. Each of the abovementioned categories correspond with values (“a” values as shown above), which are assigned prior to the use of the category multiplier (“w” values). The reasons for applying the various “a” values are shown in the second set of parentheses following each category name. For example, in category 1,  $a_1$  is shown to be either 1 or 0. If the fault code does not correspond with a voltage issue (i.e., “No” in the second set of parentheses), the “ $a_1$ ” value is 0. In another example, in category 3, if the fault code corresponds with a “component” condition, the “ $a_3$ ” value is 7. The explanation of the categories and assignment of “a” values is more fully described below. Categories 5 and 7 do not have values listed after them. Categories 5 and 7 are dependent on the characteristics of the fault code number (e.g., if a fault code has 4 interactions, then  $a_5 * w_5$  corresponds with  $4 * 1$ ). The value assignments may be greater than, less than, or just not used in other embodiments depending on the application. As such, in addition to customizing the weighting system (i.e., the multiplier values) and the categories used, a user may also edit the “a” values within each category. In summary, the priority number determination is based on the weighting of categories and the values used for each category (a two-part process). As mentioned above, the weight of categories, the categories used, and the value assignments may all be customizable based on the application.

[0051] Referring to the hierarchical categories separately, category 1—voltage—refers to the generated fault codes that indicate the engine has a voltage issue. As seen above, if the fault code represents a voltage issue,  $a_1$  is assigned a value of 1, otherwise  $a_1$  is assigned a value of 0. As most of the sensors 340 convert the information detected into a voltage read by the OBD system 330 to generate the fault codes, the accuracy of the fault codes may be affected if there is a voltage malfunction. Accordingly, voltage-related fault codes are weighted heaviest ( $w_1=10,000$ ) in the priority determination. Category 2—ECU—refers to fault codes related to the electronic control unit. If the fault code is related to the ECU,  $a_2$  is assigned a value of 1, otherwise  $a_2$  is assigned a value of 0. As the ECU controls most functions of the engine, ECU-related fault codes are weighted next heaviest after voltage-related fault codes in the priority



number determination ( $w_2=10,000$ ). In category 3, the fault code is classified into one of three areas corresponding to three different  $a_3$  values. The classes are: component ( $a_3=7$ ), system ( $a_3=3$ ), or effect ( $a_3=0$ ). A “component” fault code refers to a specific component of a system that may be malfunctioning or not within predetermined operating ranges (e.g., an exhaust gas recirculation valve is a component of the exhaust gas recirculation system). An “effect” fault code results from another failure and is typically not the root cause of the failure. Because a component of the system may be readily identified and addressed, the component fault code is assigned a larger  $a_3$  value than a system or effect  $a_3$  value. Moreover, because the fault code analyzer is identifying the root cause, a fault code designated as an “effect” fault code within category 3 is likely not to be a root cause and is weighted least heavily within the three designation types (i.e.,  $a_3=0$  for “effect” designated fault codes). After the assignment of the  $a_3$  value, the category 3 multiplier is applied ( $w_3=100$ ). Accordingly, category 3 is weighted less than categories 1 and 2.

**[0052]** Referring to category 4, the fault code is categorized based on the failure type it represents and is assigned a corresponding  $a_4$  value. According to one embodiment, within category 4, the failure types include sensor supply ( $a_4=9$ ), circuitry out-of-range ( $a_4=7$ ), smart devices ( $a_4=3$ ), and a general all other types class ( $a_4=1$ ). As seen, if the fault code is related to a sensor malfunction (“sensor supply”), the fault code is typically weighted heavier than all other classifications within category 4. Sensor malfunctions may continue to generate fault codes and, for example, warning light indicators, such that these failure types need to be addressed first. Moreover, accuracy of the component and system readings within the engine may be called into question if there is a sensor supply issue. Circuitry out-of-range denotes fault codes that indicate a specific circuit (e.g., diesel exhaust tank fluid level sensor circuit) is generating an output that is out of range of expected values. Smart devices denotes fault codes that denote possible conditions with various smart devices. After assignment of the  $a_4$  value, the multiplier ( $w_4$ ) is applied, which is, according to one embodiment, weighted less than categories 1-3 (i.e.,  $w_4=10$ ).

**[0053]** The fifth category, number interactions, refers to the number of interactions for a specific fault code in relation to the other generated fault codes. The number of interactions for a specific fault code corresponds to the  $a_5$  value. Generally, interaction fault codes are generated codes that may be related (i.e., interact) to a specific generated fault code. The term “specific fault code” refers to the fault code undergoing a priority number determination. The greater the number of interactions equates to a greater priority number impact (i.e., greater priority number). An interaction occurs if: a causal relationship between a generated fault code and the specific fault code exists (e.g., some fault codes trigger other fault codes or disable other fault codes); the specific fault code and a generated fault code relate to the same component or system; and/or the specific fault code and a generated fault code only differ based on a gradation of severity (e.g., if the specific fault code denotes a severe level and the other fault code denotes a moderately severe level of the same failure type). As an example of interaction fault codes based on a gradation of severity relationship, suppose the fault code analyzer receives five fault codes: 1673, 3547, 3498, 3497, and 3714. The corresponding descriptions of the fault codes are:

**[0054]** Fault code 1673: Aftertreatment 1 Diesel Exhaust Fluid Tank Level—Data valid but below normal operating range—Most severe level;

**[0055]** Fault code 3547: Aftertreatment Diesel Exhaust Fluid Tank Empty—Conditions exists;

**[0056]** Fault code 3498: Aftertreatment 1 Diesel Exhaust Fluid Tank Level—Data valid but below normal operating range—Moderately severe level;

**[0057]** Fault code 3497: Aftertreatment 1 Diesel Exhaust Fluid Tank Level—Data valid but below normal operating range—Least severe level; and

**[0058]** Fault code 3714: Engine Protection Derate—Condition exists.

**[0059]** Thus, fault codes 3547, 3497, 3498, and 1673 are gradations of severity for the Diesel Exhaust Fluid tank level. Accordingly, 3547, 3497, and 3498 are “interactions” of fault code 1673 (therefore, fault code 1673 has 3 interactions: 3547, 3497, and 3498). However, from the above relationship factors, fault code 3714 has zero interactions. If this example was carried through method **200** (or method **300** below), fault code 1673 would generate the highest priority number and be identified as a root cause based on its number of interactions and it being the designated as the most severe code. Accordingly, a mechanic or technician would address fault code 1673 first. Based on the interactions, troubleshooting 1673 may resolve the malfunction root cause and clear the other interaction fault codes. In sum, the number of interactions is determined by the fault code analyzer and the higher the number of interactions, the greater the effect there is on the priority number determination. However, the weighting applied to category 5 ( $w_5$ ) is less than that of categories 1-4 ( $w_5=1$ ), such that category 5 has less of an impact on the determined priority number than categories 1-4.

**[0060]** Still referring to the fault code hierarchy categories, category 6 refers to the location, which is sub-classified as engine or aftertreatment, according to one configuration. If the fault code is related to the engine, the  $a_6$  value is 1, otherwise the  $a_6$  value (for aftertreatment) is 0. In operation, engine fault codes are more likely to cause aftertreatment fault codes than vice versa. Accordingly, a greater emphasis is placed on engine fault codes than on after treatment fault codes. For example, aftertreatment fault codes may refer to possible malfunctions in the emissions system. According to one embodiment, the aftertreatment system treats exhaust gas from a diesel engine to reduce the presence of nitrogen oxides, other pollutants, and unwanted particles in the exhaust gas. Engine fault codes relate to the operation of the engine (i.e., pre-aftertreatment). According to alternate embodiments, category 6 may include other location categories, such as a braking system with a specific “ $a_6$ ” value for each class within category 6.

**[0061]** The last and least weighted category ( $w_7=0.01$ ) in the priority number determination is category 7, fault code number. In some embodiments,  $a_7$  is determined by dividing the fault code number by one-thousand (e.g.,  $a_7=1.673=1673/1000$ ). According to one embodiment, category 7 may be utilized as a tie-breaker for priority number determinations. Since all fault codes are assigned different fault code numbers, the insertion of this category in the priority number determination will prevent a tie in priority numbers.

**[0062]** With the above in mind, an example priority number determination is shown in FIG. 6 for fault code number 1673. As mentioned above, fault code 1673 corresponds

with a diesel exhaust fluid tank. Accordingly, fault code 1673 is not related to a voltage issue ( $a_1$  is 0); is not related to the ECM ( $a_2$  is 0); is related to a specific component—the diesel exhaust fluid tank ( $a_3$  is 7); is not classified into a sensor supply, circuitry out-of-range, or smart device category ( $a_4$  is 1); using the above example as mentioned above, has 4 interactions ( $a_5$  is 4); is not related to the engine ( $a_6$  is 0); and has an  $a_7$  value of 1.673 (i.e., 1673/1000). Thus, the priority number for fault code 1673 is determined as follows:

$$714.01673 = (0)(10,000) + (0)(1,000) + (7)(100) + (1)(10) + (4)(1) + (0)(0.1) + (1.673)(0.01)$$

**[0063]** This example determination may be used for each fault code in order to prioritize all of the fault codes (based on the highest priority number). In some embodiment, the memory 354 of the user device 350 and/or the memory 364 of the server 360 store a look-up table that maps each fault code to a corresponding priority numbers.

**[0064]** After each fault code is prioritized, the fault code analyzer determines the first root cause. In certain embodiments, the first root cause is the fault code number having the highest priority rating (“number”). The fault code analyzer may then remove this fault code number (first root cause) and its associated interaction fault codes from the original list of fault codes. As such, one or more unused fault codes remain. The fault code analyzer may determine that the next highest priority fault code (from the fault codes remaining) is the second root cause and determine its associated interaction fault codes. This process of identifying root cause fault numbers and associated interaction fault numbers may continue until all identified fault codes are accounted for as either a root cause, interaction fault code, or separate fault code. In some embodiments, troubleshooting the first root cause may clear the remaining fault codes. In some embodiments, up to three root cause fault codes are identified from the fault codes received.

**[0065]** Referring back to FIG. 4, the labor time determining circuit 406 is structured to determine an estimated labor time for fixing the root cause identified by the root cause determining circuit 404. In some embodiments, the memory 354 of the user device 350 and/or the memory 364 of the server 360 store, for example, a look-up table that maps fault codes to corresponding labor hours. The labor time determining circuit 406 uses the look-up table to determine the estimated labor time. In some embodiments, the user device 350 and/or the server 360 use the identified root cause fault code to query the database 368 and determines the estimated labor hours based on the returned results.

**[0066]** Referring to FIG. 7, a flow diagram of a method 700 for determining estimated labor hours is shown according to an example embodiment. The method 700 is implemented on the server 360 (e.g., executed by the processor 362). At a process 702, historical data of labor hours are collected. As discussed above, the server 360 can receive information relating to warranty claims from service providers around the world via the network 304. As used herein, a warranty claim refers to a claim for reimbursement of material, labor, and administration costs that are incurred for the purpose of rectifying faults in a product with a valid warranty. A certified service provider can upload, for example, a warranty claim form to the server 360 or fill the form online at a website associated with the server 360. The warranty claim form may include various data such as engine serial number (ESN), year, make, model, adminis-

tration time, repair time, diagnostic session identifier (DSID), and so on. As used herein, a DSID refers to an identifier created for a diagnostic session that uniquely identifies the diagnostic session. The server 360 can extract the data from the warranty claim form and store the data in the database 368. In some embodiments, warranty claims are provided as unstructured data to the server 360. The server 360 can extract some or all of the above data from the text of the warranty claims. In some embodiments, the server 360 can also collect data of historical labor hours out of the context of warranty claims. For example, a service center of a manufacture can record information relating to repair performed at the center (e.g., ESN, year, make, model, administration time, repair time, DSID) for the purpose of data mining.

**[0067]** At a process 704, historical data of labor time associated with the determined root cause(s) are sorted out. To develop and improve knowledge base for automatic troubleshooting, the server 360 can receive repair information from service providers around the world. In some embodiments, a feedback system is provided at the server 360 for continuous improvement of the troubleshooting knowledge base. In particular, when conducting diagnosis, a service technician creates a diagnostic session identified by the DSID with the feedback system, and inputs the initial complaint, the technician’s assessment of symptoms, and any fault codes reported by the OBD to the feedback system. The ESN, year, make, model, etc. can also be input to the feedback system. Based on the information, the feedback system maps the complaint to standard troubleshooting terms and may present further guided questions to the technician in an optimized order. After obtaining the technician’s answers to the guided questions, the system provides a list of candidate solutions based on the knowledge base in an order of likelihood and troubleshooting logic. Each solution points to a procedure in the knowledge base. The technician enters the verification results when going through each solution. If the technician finds a solution out of the list of candidate solutions, the feedback system documents the repair procedure and how the repair has been performed, and updates the knowledge base with the new repair procedure.

**[0068]** In some embodiments, the warranty claims, which include labor time, and the repair information, which includes fault codes, are stored as separate pieces of data in the database 368. The server 360 can connect the warranty claims and the repair information by using, for example, the DSID. For example, the server 360 uses the determined root cause fault code(s) to query the database, which returns the repair information relating to the root cause. The server 360 then uses the DSID included in the repair information to query the database 368, which returns the warranty claims relating to the DSID. Thus, historical labor time associated with the root cause are sorted out. In further embodiments, the server 360 can subtract the administration time from the repair time to obtain the labor time. In some embodiments, the server 360 excludes the historical data if the DSID and/or ESN are not valid, and/or the make/year/model is unknown. In some embodiments in which multiple root causes are present, the sever 360 can sort out the historical data of labor time associated with each root cause.

**[0069]** At a process 706, an estimated labor time is determined based on statistics of history data of labor time associated with the determined root cause. In some embodi-

ments, the server **360** picks a size (e.g., 5, 10, 15, 20, etc.) of samples from the historical data of labor time associated with the root cause, and determines the labor time to be a predefined percentile for the historical labor time. The size and/or pool of sampling can vary for different root causes. For example, for a fault code with high volume (i.e., a fault code that occurs at a high frequency), a minimum of 20 pieces of labor time data for repairs that happened within one month prior to date are sampled. In some embodiments, the make/year/model of the historical data are matched with the vehicle **302** being examined. In some embodiments, at least some of make/year/model are not matched. As another example, for a fault code with medium volume (i.e., a fault code that occurs at a medium frequency), a minimum 10 pieces of labor hours data for repairs that happened within three months prior to date are sampled. For a fault code with low volume (i.e., a fault code that occurs at a low frequency), a minimum of 5 pieces of labor hours data collected all through are sampled. Then the estimated labor time can be determined to be a value that is higher than 75% of the sampled labor hours but not higher than the rest 25% of the sampled labor hours. It shall be noted that the percentile 75% is given for illustration not for limitation. Other suitable percentile can also be used, such as 60%, 65%, 70%, 80%, and so on. In some embodiments in which multiple root causes are identified, the estimated labor time can be a sum of estimated labor time for each root cause.

**[0070]** Referring back to FIG. 4, the assessment output circuit **408** is structured to output the root cause(s) and the estimated labor time to a display. The display can be the user interface **357** (e.g., a touchscreen) of the user **350**. In some embodiments in which the assessment output circuit **408** is implemented on the user device **350**, the root cause(s) and the estimated labor time are output to the user interface **357** through an internal bus. In some embodiments in which the assessment output circuit **408** is implemented on the server **360**, the root cause(s) and the estimated labor time are output to the user interface **357** via the network **304**. In some embodiments, the assessment output circuit **408** can further output other information together with the root cause(s) and the estimated labor time, such as description for the root cause fault code, name of the component that possibly broke, and/or the category of difficulty of diagnosing and repairing (e.g., “easy,” “hard,” “normal”).

**[0071]** Referring now to FIG. 8, a flow diagram of a method **800** for assessing a service event associated with a vehicle is shown according to an example embodiment. The method **800** may be implemented with the user device **350**, the server **360**, or a combination thereof.

**[0072]** At a process **802**, at least one fault code of an ECU is received. In some embodiments, the fault code(s) are received from the diagnostic port **332** through the adapter **306**. In some embodiments, the fault code(s) are received from the telematics device **320** on the vehicle **302**. In some embodiments, other vehicle status and conditions information is received together with the fault code(s), such as VIN, make, model, year and month of production, serial number, mileage, engine speed, etc.

**[0073]** At a process **804**, at least one root cause is determined based on the at least one fault code. In some embodiments, the root cause is represented by one fault code from multiple fault codes received from the vehicle. In some embodiment, the root cause is represented by the name of a component, the failure of which might have caused the

received fault code(s). In some embodiments, a fault code analyzer as described in the U.S. patent application Ser. No. 14/219,755 is used where an engine has generated multiple fault codes. The fault code analyzer can put the multiple fault codes in sequence and point to up to three root cause fault codes. In particular, when multiple fault codes are received, the fault code analyzer categorizes what the fault code relates to (e.g., voltage, ECM, etc.) and determines a priority for each fault code. Typically, the priority ranking is based on a priority number. The higher the priority number, the more likely that fault code is the root cause of the problem. The fault code with the highest priority number may be designated as the first root cause. The fault code analyzer may list any and all interaction fault codes for the first root cause fault code. Determining that a fault code is an interaction fault code may be based on many characteristics (e.g., each fault code is related to the same engine or vehicle component). If any fault codes remain, the fault code analyzer will designate a second root cause fault code, based on the next highest priority number for the unaccounted fault codes and provide the remaining interaction fault codes for the second root cause. In some embodiments, the fault code analyzer analyzes inactive fault codes in addition to active fault codes, which increases the possibility of identifying the correct and all root causes. In some embodiments, troubleshooting the first root cause may clear the remaining fault codes. In some embodiments, up to three root cause fault codes are identified from the fault codes received. In some embodiments, a look-up table that maps fault codes to priority numbers is stored in a memory. The fault code with the highest priority number is identified as the root cause by referring to the look-up table.

**[0074]** At a process **806**, an estimated labor time is determined for the at least one root cause. In some embodiments, determining the estimated labor time includes collecting historical data of labor hours, sorting out historical data of labor hours associated with the identified root cause(s), and determining an estimated labor time based on statistics of history data of labor time associated with the identified root cause. For example, information relating to warranty claims are collected from service providers around the world. The warranty claims may include various data such as ESN, year, make, model, administration time, repair time, diagnostic session identifier DSID, and so on. Historical data of labor hours can also be collected from sources out of the context of warranty claims in some embodiments.

**[0075]** Repair information can also be collected from service providers around the world. In some embodiments, the repair information is collected through a feedback system provided for continuous improvement of the troubleshooting knowledge base. In particular, when conducting diagnosis, a service technician creates a diagnostic session identified by the DSID with the feedback system, and inputs the initial complaint, the technician's assessment of symptoms, and any fault codes reported by the OBD to the feedback system. The ESN, year, make, model, etc. can also be input to the feedback system. Based on the information, the feedback system maps the complaint to standard troubleshooting terms and may present further guided questions to the technician in an optimized order. After obtaining the technician's answers to the guided questions, the system provides a list of candidate solutions based on the knowledge base in an order of likelihood and troubleshooting logic. Each solution points to a procedure in the knowledge

base. The technician enters the verification results when going through each solution. If the technician finds a solution out of the list of candidate solutions, the feedback system documents the repair procedure and how the repair has been performed, and updates the knowledge base with the new repair procedure. In some embodiments, the warranty claims, which include labor time information, and the repair information, which includes fault code information, are connected by using the DSID. Thus, historical data of the labor time associated with the root cause fault code(s) are sorted out.

**[0076]** In some embodiments, a size (e.g., 5, 10, 15, 20) of samples are picked from the historical data of labor time associated with the root cause, and the labor time is determined to be a predefined percentile for the historical labor time. The size and/or pool of sampling can vary for different root causes. For example, for a fault code with high volume (i.e., a fault code that occurs at a high frequency), a minimum of 20 pieces of labor hours data for repairs that happened within one month prior to date are sampled. In some embodiments, the make/year/model of the historical data are matched with the vehicle being examined. In some embodiments, at least some of make/year/model are not matched. As another example, for a fault code with medium volume (i.e., a fault code that occurs at a medium frequency), a minimum 10 pieces of labor hours data for repairs that happened within three months prior to date are sampled. For a fault code with low volume (i.e., a fault code that occurs at a low frequency), a minimum of 5 pieces of labor hours data collected all through are sampled. Then the labor time can be determined to be a value that is higher than 75% of the sampled labor hours but not higher than the rest 25% of the sampled labor hours. Other suitable percentile can also be used, such as 60%, 65%, 70%, 80%, and so on. In some embodiments where more than one root cause are identified, the estimated labor time can be a sum of estimated labor time for each root cause.

**[0077]** At a process **808**, the root cause(s) and the estimated labor time are output to a display. The display may be a screen of a user device (e.g., smartphone, tablet). In some embodiments, other information can be output together with the root cause(s) and the estimated time to the display, such as description for the root cause fault code, name of the component that possibly breaks, and/or the category of difficulty of diagnosing and repairing (e.g., “easy,” “hard,” “normal”). The displayed information can be used for scheduling the service event for the vehicle, as described above in reference to FIGS. 1(A)-1(C) and 2(A)-2(B).

**[0078]** It should be understood that no claim element herein is to be construed under the provisions of 35 U.S.C. §112(f), unless the element is expressly recited using the phrase “means for.” The schematic flow chart diagrams and method schematic diagrams described above are generally set forth as logical flow chart diagrams. As such, the depicted order and labeled steps are indicative of representative embodiments. Other steps, orderings and methods may be conceived that are equivalent in function, logic, or effect to one or more steps, or portions thereof, of the methods illustrated in the schematic diagrams. Further, reference throughout this specification to “one embodiment,” “an embodiment,” “an example embodiment,” or similar language means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the present

invention. Thus, appearances of the phrases “in one embodiment,” “in an embodiment,” “in an example embodiment,” and similar language throughout this specification may, but do not necessarily, all refer to the same embodiment.

**[0079]** Additionally, the format and symbols employed are provided to explain the logical steps of the schematic diagrams and are understood not to limit the scope of the methods illustrated by the diagrams. Although various arrow types and line types may be employed in the schematic diagrams, they are understood not to limit the scope of the corresponding methods. Indeed, some arrows or other connectors may be used to indicate only the logical flow of a method. For instance, an arrow may indicate a waiting or monitoring period of unspecified duration between enumerated steps of a depicted method. Additionally, the order in which a particular method occurs may or may not strictly adhere to the order of the corresponding steps shown. It will also be noted that each block of the block diagrams and/or flowchart diagrams, and combinations of blocks in the block diagrams and/or flowchart diagrams, can be implemented by special purpose hardware-based systems that perform the specified functions or acts, or combinations of special purpose hardware and program code.

**[0080]** Many of the functional units described in this specification have been labeled as circuits, in order to more particularly emphasize their implementation independence. For example, a circuit may be implemented as a hardware circuit comprising custom very-large-scale integration (VLSI) circuits or gate arrays, off-the-shelf semiconductors such as logic chips, transistors, or other discrete components. A circuit may also be implemented in programmable hardware devices such as field programmable gate arrays, programmable array logic, programmable logic devices or the like.

**[0081]** As mentioned above, circuits may also be implemented in machine-readable medium for execution by various types of processors, such as the processor **400** of FIG. 4. An identified circuit of executable code may, for instance, comprise one or more physical or logical blocks of computer instructions, which may, for instance, be organized as an object, procedure, or function. Nevertheless, the executables of an identified circuit need not be physically located together, but may comprise disparate instructions stored in different locations which, when joined logically together, comprise the circuit and achieve the stated purpose for the circuit. Indeed, a circuit of computer readable program code may be a single instruction, or many instructions, and may even be distributed over several different code segments, among different programs, and across several memory devices. Similarly, operational data may be identified and illustrated herein within circuits, and may be embodied in any suitable form and organized within any suitable type of data structure. The operational data may be collected as a single data set, or may be distributed over different locations including over different storage devices, and may exist, at least partially, merely as electronic signals on a system or network.

**[0082]** The computer readable medium (also referred to herein as machine-readable media or machine-readable content) may be a tangible computer readable storage medium storing the computer readable program code. The computer readable storage medium may be, for example, but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, holographic, micromechanical, or semiconductor

system, apparatus, or device, or any suitable combination of the foregoing. As alluded to above, examples of the computer readable storage medium may include but are not limited to a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), a portable compact disc read-only memory (CD-ROM), a digital versatile disc (DVD), an optical storage device, a magnetic storage device, a holographic storage medium, a micromechanical storage device, or any suitable combination of the foregoing. In the context of this document, a computer readable storage medium may be any tangible medium that can contain, and/or store computer readable program code for use by and/or in connection with an instruction execution system, apparatus, or device.

**[0083]** Computer readable program code for carrying out operations for aspects of the present invention may be written in any combination of one or more programming languages, including an object oriented programming language such as Java, Smalltalk, C++ or the like and conventional procedural programming languages, such as the “C” programming language or similar programming languages.

**[0084]** The program code may also be stored in a computer readable medium that can direct a computer, other programmable data processing apparatus, or other devices to function in a particular manner, such that the instructions stored in the computer readable medium produce an article of manufacture including instructions which implement the function/act specified in the schematic flowchart diagrams and/or schematic block diagrams block or blocks.

**[0085]** Accordingly, the present disclosure may be embodied in other specific forms without departing from its spirit or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. The scope of the disclosure is, therefore, indicated by the appended claims rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.

What is claimed is:

1. A system for service assessment, the system comprising:

- an input circuit structured to receive at least one fault code of an engine control unit (ECU);
- a root cause determination circuit structured to determine at least one root cause based on the at least one fault code of the ECU;
- a labor time estimation circuit structured to determine an estimated labor time for fixing the at least one root cause; and
- an output circuit structured to output the at least one root cause and the estimated labor time to a display.

2. The system of claim 1, wherein the input circuit is structured to receive the at least one fault code from an on-board diagnostic (OBD) system coupled to the ECU.

3. The system of claim 1, further comprising a fault code analyzing circuit structured to analyze more than one fault codes and prioritize the more than one fault codes.

4. The system of claim 1, wherein multiple fault codes are received, and wherein the root cause determination circuit is structured to:

- assign a priority number to each of the multiple fault codes; and

designate a fault code with the highest priority number as the root cause.

5. The system of claim 1, wherein the labor time estimation circuit is structured to:

- collect historical data of labor hours;
- sort out history data of labor time associated with the at least one root cause; and
- estimate the labor time by statistically analyzing the historical labor time data associated with the at least one root cause.

6. The system of claim 5, wherein the historical labor time data includes data collected from historical warranty claims.

7. The system of claim 5, wherein the estimated labor time is at a predefined percentile for the historical labor time data associated with the at least one root cause.

8. A method for service assessment, the method comprising:

- receiving at least one fault code of an engine control unit (ECU);
- determining at least one root cause based on the at least one fault code of the ECU;
- determining an estimated labor time for fixing the at least one root cause; and
- outputting the at least one root cause and the estimated labor time to a display.

9. The method of claim 8, further comprising scheduling a service event to fix the at least one root cause based on the at least one root cause and the estimated labor time.

10. The method of claim 8, further comprising analyzing more than one fault codes and prioritizing the more than one fault codes.

11. The method of claim 8, wherein the determining the at least one root cause includes:

- assigning a priority number to each of the multiple fault codes; and
- designating a fault code with the highest priority number as the root cause.

12. The method of claim 8, wherein estimating the labor time includes:

- collecting historical data of labor hours;
- sorting out history data of labor time associated with the at least one root cause; and
- estimating the labor time by statistically analyzing the historical labor time data associated with the at least one root cause.

13. The method of claim 12, wherein the estimated labor time is at a predefined percentile for the historical labor time data associated with the at least one root cause.

14. A user device comprising:

- an input port structured to receive at least one fault code of an engine control unit (ECU);
- a controller coupled to the input port, the controller structured to:
  - determine at least one root cause based on the at least one fault code; and
  - determine an estimated labor time for fixing the at least one root cause; and
- a display coupled to the controller, the display structured to display the at least one root cause and the estimated labor time.

15. The user device of claim 14, wherein the input port is structured to receive the at least one fault code from the on-board diagnostic (OBD) system coupled to the ECU.

**16.** The user device of claim **14**, wherein the controller is structured to analyze more than one fault codes and prioritize the more than one fault codes.

**17.** The user device of claim **14**, further comprising a memory storing a look-up table that maps fault codes to priority numbers, wherein the controller is structured to designate a fault code with the highest priority number as the root cause.

**18.** The user device of claim **14**, further comprising a memory storing a look-up table that maps fault codes to labor hours, wherein the controller is structured to determine the estimated labor time based on the look-up table.

**19.** The user device of claim **14**, further comprising a transceiver structured to transmit the at least one fault code to a remote server, and receive the at least one root cause and the estimated labor time from the remote server.

**20.** The user device of claim **14**, wherein the user device includes a smartphone or a tablet.

\* \* \* \* \*