

(12) Indian Patent Application

(21) Application Number: 202441030121

(22) Filing Date: 12/04/2024 (43) Publication Date: 17/10/2025

(71) Applicant(s): L&T TECHNOLOGY SERVICES LIMITED

(72) Inventor(s): Sureshbabu, Niveditha
Raj, Rajesh
Singh, Madhusudan

(51) International Classifications: G06N 3/08 G06F 40/20 G06F 11/36

(54) Title: METHOD AND SYSTEM OF DETERMINING TEST PROCEDURES USING LARGE LANGUAGE MODELS

(57) Abstract: A method (300) and system (100) of determining test procedures using large language models (LLMs) is disclosed. A processor (104) receives a plurality of domain-based documents corresponding to a test product to be tested. One or more user requirements are determined corresponding to at least one feature of the test product from one of the plurality of domain-based documents. A plurality of domain-specific keywords is determined from the one or more user requirements. A contextual data is determined by extracting a portion of a text data from the plurality of domain-based documents. A knowledge dataset is determined by prompting a second LLM based on a second prompt and the contextual data. One or more test procedures for one or more test cases for testing the at least one feature of the test product by prompting a third LLM based on a third prompt and the knowledge dataset.

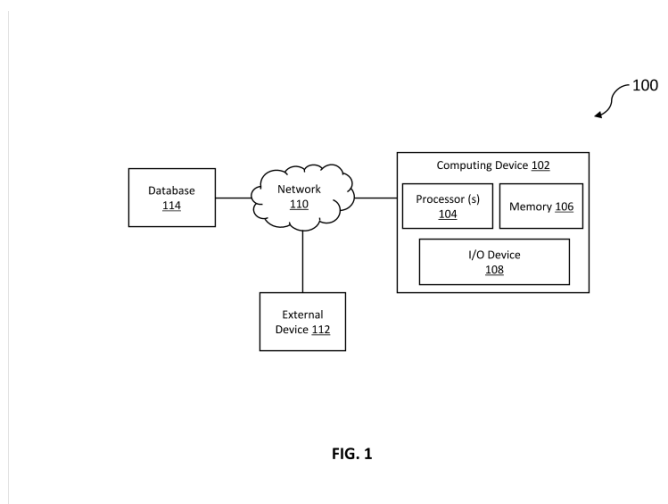


FIG. 1

FORM 2

THE PATENTS ACT 1970
(39 OF 1970)

&

The Patent Rules, 2003

Complete Specification

(See Section 10 and Rule 13)

1. TITLE OF THE INVENTION

METHOD AND SYSTEM OF DETERMINING TEST PROCEDURES USING LARGE LANGUAGE MODELS

2. APPLICANT(S)

(a) NAME : **L&T TECHNOLOGY SERVICES LIMITED**

(b) NATIONALITY : **INDIAN**

(c) ADDRESS : **DLF IT SEZ Park, 2nd Floor – Block 3**

1/124, Mount Poonamallee Road,

Ramapuram, Chennai – 600 089,

INDIA.

3. PREAMBLE TO THE DESCRIPTION

COMPLETE

The following specification particularly describes the invention and the manner in which it is
to be performed

DESCRIPTION

Technical Field

[001] This disclosure relates generally to content generation and more particularly to a method and system of determining content generation using Large Language Models (LLMs).

5

BACKGROUND

[002] Testing is critical process when it comes to development of new software, electrical component, electronics components, etc. The current testing processes rely heavily on manual generation of test cases and test procedures. In order to create test cases and test procedures, testers are required to have sufficient domain-knowledge to test all necessary features of a product. This manual approach may consume significant time in creation of the test cases. Also, lack of sufficient domain knowledge may lead to creation of non-exhaustive test cases and procedures. Automation in testing also fails to provide an efficient solution to create exhaustive test cases and test procedure to test all the technical features of a product due to lack of domain knowledge.

15

[003] Therefore, there is a requirement for an efficient and effective methodology for determining test procedures based on domain knowledge.

SUMMARY OF THE INVENTION

[004] In an embodiment, a method for determining test procedures using large language models (LLMs) is disclosed. The method may include receiving, by a processor, a plurality of domain-based documents corresponding to a test product to be tested. The method may further include determining, by the processor, one or more user requirements corresponding to at least one feature of the test product from one of the plurality of domain-based documents. The method may further include determining, by the processor, a plurality of domain-specific keywords from the one or more user requirements by prompting a first LLM based on a first prompt. The method may further include determining, by the processor, contextual data by extracting a portion of text data from the plurality of domain-based documents. In an embodiment, the portion of the text data may include one or more of the plurality of domain-specific keywords. The method may further include determining by the processor, a knowledge dataset by prompting a second LLM based on a second prompt and the contextual data. In an embodiment, the knowledge dataset may include a set of questions and a set of answers to each of the set of

20

25

30

questions based on the contextual data and the text data of the plurality of domain-based documents. In an embodiment, the second prompt may be engineered to prompt the second LLM to list the set of questions and the set of answers to each of the set of questions based on the contextual data and the text data of the plurality of domain-based documents. The method
5 may further include determining, by the processor, one or more test procedures for one or more test cases for testing the at least one feature of the test product by prompting a third LLM based on a third prompt and the knowledge dataset. In an embodiment, the third prompt may include the one or more test cases for testing the at least one feature of the test product.

[005] In another embodiment, a system of determining test procedures based on large language
10 models (LLMs) is disclosed. The system may include a processor, a memory communicably coupled to the processor, wherein the memory may store processor-executable instructions, which when executed by the processor may cause the processor to receive a plurality of domain-based documents corresponding to a test product to be tested. The processor may further determine one or more user requirements corresponding to at least one feature of the
15 test product from one of the plurality of domain-based documents. The processor may further determine a plurality of domain-specific keywords from the one or more user requirements by prompting a first LLM based on a first prompt. The processor may further determine a contextual data by extracting a portion of text data from the plurality of domain-based documents. In an embodiment, the portion of the text data may include one or more of the
20 plurality of domain-specific keywords. The processor may further determine a knowledge dataset by prompting a second LLM based on a second prompt and the contextual data. In an embodiment, the knowledge dataset may include a set of questions and a set of answers to each of the set of questions based on the contextual data and the text data of the plurality of domain-based documents. In an embodiment, the second prompt may be engineered to prompt the
25 second LLM to list the set of questions and the set of answers to each of the set of questions based on the contextual data and the text data of the plurality of domain-based documents. The processor may further determine one or more test procedures for one or more test cases for testing the at least one feature of the test product by prompting a third LLM based on a third prompt and the knowledge dataset. In an embodiment, the third prompt may include the one or
30 more test cases for testing the at least one feature of the test product.

[006] It is to be understood that both the foregoing general description and the following detailed descriptions are exemplary and explanatory only and are not restrictive of the invention, as claimed.

BRIEF DESCRIPTION OF THE DRAWING

[007] FIG. 1 illustrates a block diagram of an exemplary test procedures determination system for determining test procedures using large language models, in accordance with an embodiment of the present disclosure.

5 [008] FIG. 2 illustrates a functional block diagram of a computing device, in accordance with an embodiment of the present disclosure.

[009] FIG. 3 illustrates a flow diagram of a method of determining test procedures using large language models, in accordance with an embodiment of present disclosure.

DETAILED DESCRIPTION OF THE DRAWINGS

10 [010] Exemplary embodiments are described with reference to the accompanying drawings. Wherever convenient, the same reference numbers are used throughout the drawings to refer to the same or like parts. While examples and features of disclosed principles are described herein, modifications, adaptations, and other implementations are possible without departing from the scope of the disclosed embodiments. It is intended that the following detailed
15 description be considered exemplary only, with the true scope being indicated by the following claims. Additional illustrative embodiments are listed.

[011] Further, the phrases “in some embodiments”, “in accordance with some embodiments”, “in the embodiments shown”, “in other embodiments”, and the like mean a particular feature, structure, or characteristic following the phrase is included in at least one embodiment of the
20 present disclosure and may be included in more than one embodiment. In addition, such phrases do not necessarily refer to the same embodiments or different embodiments. It is intended that the following detailed description be considered exemplary only, with the true scope being indicated by the following claims.

[012] Large Language Models (LLMs) are being leveraged for performing various authoring
25 tasks due to their text generation capacity. However, due to poor prompting language sometimes LLMs provide irrelevant and incorrect output. Further, LLMs may also rely on the generic training data that may not be sufficient to be able for them to provide output related to niche domains of the test product.

[013] In order to automate the test case creation process, LLMs may be required to be trained up to a level of a domain expert. Such training necessitates the infusion of domain and external knowledge. The present disclosure provides a system and a method of determining test procedures using LLMs.

5 [014] Referring now to FIG. 1, a block diagram of an exemplary test procedures determination system 100 is illustrated, in accordance with an embodiment of the present disclosure. The test procedure determination system 100 may include a computing device 102, an external device 112, and a database 114 communicably coupled to each other through a wired or wireless communication network 110. The computing device 102 may include a processor 104, a
10 memory 106 and an input/output (I/O) device 108.

[015] In an embodiment, examples of processor(s) 104 may include, but are not limited to, an Intel® Itanium® or Itanium 2 processor(s), or AMD® Opteron® or Athlon MP® processor(s), Motorola® lines of processors, Nvidia®, FortiSOC™ system on a chip processors or other future processors.

15 [016] In an embodiment, the memory 106 may store instructions that, when executed by the processor 104, may cause the processor 104 to determine test procedures using a plurality of LLMs, as discussed in more detail below. In an embodiment, the memory 106 may be a non-volatile memory or a volatile memory. Examples of non-volatile memory may include but are not limited to, a flash memory, a Read Only Memory (ROM), a Programmable ROM (PROM),
20 Erasable PROM (EPROM), and Electrically EPROM (EEPROM) memory. Further, examples of volatile memory may include but are not limited to, Dynamic Random Access Memory (DRAM), and Static Random-Access memory (SRAM).

[017] In an embodiment, the I/O device 108 may comprise of variety of interface(s), for example, interfaces for data input and output devices, and the like. The I/O device 108 may
25 facilitate inputting of instructions by a user communicating with the computing device 102. In an embodiment, the I/O device 108 may be wirelessly connected to the computing device 102 through wireless network interfaces such as Bluetooth®, infrared, or any other wireless radio communication known in the art. In an embodiment, the I/O device 108 may be connected to
30 a communication pathway for one or more components of the computing device 102 to facilitate the transmission of inputted instructions and output results of data generated by various components such as, but not limited to, the processor(s) 104 and data saved in the memory 106.

[018] In an embodiment, the database 114 may be enabled in a cloud or a physical database and may store a plurality of domain-based documents, knowledge dataset, and contextual data. In an embodiment, the database 114 may store data input by an external device 112 or output generated by the computing device 102. In an embodiment, the domain-based documents may include domain based technical information related to a test product to be tested. In an exemplary embodiment, the domain-based documents may include architecture, specifications, CAN matrix and so on in case of an electrical test product.

[019] In an embodiment, the communication network 110 may be a wired or a wireless network or a combination thereof. The network 110 can be implemented as one of the different types of networks, such as but not limited to, ethernet IP network, intranet, local area network (LAN), wide area network (WAN), the internet, Wi-Fi, LTE network, CDMA network, 5G and the like. Further, network 110 can either be a dedicated network or a shared network. The shared network represents an association of the different types of networks that use a variety of protocols, for example, Hypertext Transfer Protocol (HTTP), Transmission Control Protocol/Internet Protocol (TCP/IP), Wireless Application Protocol (WAP), and the like, to communicate with one another. Further network 110 can include a variety of network devices, including routers, bridges, servers, computing devices, storage devices, and the like.

[020] In an embodiment, the computing device 102 may receive a request to determine test procedures using Large Language Models (LLMs) from an external device 112 through the network 110. In an embodiment, the computing device 102 and the external device 112 may be a computing system, including but not limited to, a smart phone, a laptop computer, a desktop computer, a notebook, a workstation, a portable computer, a handheld, or a mobile device. In an embodiment, the computing device 102 may be, but not limited to, in-built into the external device 112 or may be a standalone computing device.

[021] In an embodiment, the computing device 102 may perform various processing in order to determine test procedures using LLMs. In an embodiment, examples of the LLMs may include, but are not limited to, zephyr, code LLAMA, GPT, etc. By way of an example, the computing device 102 may receive a plurality of domain-based documents corresponding to a test product to be tested from a user via an I/O device 108. In an embodiment, the plurality of domain-based documents corresponding to the test product to be tested may be stored in the database 114. In an embodiment, the plurality of domain-based documents may include a documents including one or more user requirements.

[022] Accordingly, the computing device 102 may determine one or more user requirements corresponding to at least one feature of the test product from one of the plurality of domain-based documents. The computing device 102 may determine a plurality of domain-specific keywords from the one or more user requirements by prompting a first LLM using a first prompt. In an embodiment, the first prompt may be engineered to prompt the first LLM to output the plurality of domain-specific keywords by determining a set of nouns based on the one or more user requirements corresponding to the test product to be tested.

[023] Further, the computing device 102 may determine contextual data by extracting a portion of text data from the plurality of domain-based documents. In an embodiment, the portion of the text data may include one or more of the plurality of domain-specific keywords. Further, in an embodiment, the contextual data may be determined based on determination of a positional relation between each of the plurality of domain-specific keywords with the text data. In an embodiment, the positional relation may be determined based on a lookup of each of the plurality of domain-specific keywords in the text data of the plurality of domain-specific documents.

[024] Further, the computing device 102 may determine a knowledge dataset by prompting a second LLM using a second prompt and the contextual data. In an embodiment, the knowledge dataset may include a set of questions and a set of answers to each of the set of questions based on the contextual data and the text data of the plurality of domain-based documents. In an embodiment, the second prompt may be engineered to prompt the second LLM to list the set of questions and the set of answers to each of the set of questions based on the contextual data and the text data of the plurality of domain-based documents.

[025] Further, the computing device 102 may determine one or more test cases for testing the at least one feature of the test product by prompting a fourth LLM using a fourth prompt. In an embodiment, the fourth prompt may be engineered to prompt the fourth LLM to list the one or more test cases corresponding to the one or more user requirements. In an embodiment, the test cases may correspond to various scenarios to at least one user requirement to be tested corresponding to the at least one feature of the test product.

[026] Further, the computing device 102 may determine one or more test procedures for each of the one or more test cases for testing the at least one feature of the test product by prompting a third LLM using a third prompt and the knowledge dataset. In an embodiment, the third prompt may include the one or more test cases for testing the at least one feature of the test

product. In an embodiment, each of the one or more test procedures may include at least one pre-condition, a set action steps to be performed for testing the test product, and at least one expected outcome for the at least one pre-condition.

5 [027] In an embodiment, the third prompt may be determined as an output generated by the fourth LLM queried using the fourth prompt. Accordingly, the one or more test cases corresponding to the one or more user requirements may be listed by the fourth LLM as output. In an embodiment, the third prompt may be engineered to output the one or more test procedures for the one or more test cases based on the knowledge dataset and the one or more user requirements. In an embodiment, the third
10 prompt may be engineered to prompt the third LLM to output the one or more test procedures for the one or more test cases based on the knowledge dataset and the one or more user requirements.

[028] Referring now to **FIG. 2**, a functional block diagram of the computing device 102 is illustrated, in accordance with an embodiment of the present disclosure. In an embodiment, the
15 computing device 102 may receive a plurality of domain-based documents corresponding to a test product to be tested.

[029] In an embodiment, the computing device 102 may include a user requirement determination module 202, a domain-specific keywords determination module 204, a contextual data determination module 206, a knowledge dataset determination module 212, a
20 test cases determination module 214, a test procedures determination module 216.

[030] The user requirement determination module 202 may determine one or more user requirements corresponding to at least one feature of the test product from one of the plurality of domain-based documents. In an embodiment, the domain-based documents may include domain based technical information related to a test product to be tested. In an exemplary
25 embodiment, the domain-based documents may include architecture, specifications, CAN matrix and so on in case of an electrical test product.

[031] In an embodiment, one of the plurality of domain-based documents may include a user requirement document. In an embodiment, the user requirement document may include one or more user requirements for testing at least one feature of the test product. According to an
30 exemplary embodiment, an exemplary user requirement document may list one or more user requirements to test one or more features of a test product:

“1. The vehicle inlet power contacts (SAE J1772) shall be electrically isolated from the battery to avoid electric shock when the connector is removed from the vehicle inlet and detection of the proximity pin disconnection.

2. Communication and connection between OBEVC Charger and the Grid follow the SAEJ1772 connector standard.

3. When the LV battery is fully charged to 14.1V then, OBEVC shall:

a) cut-off the power supply to the LV battery

b) stop charging LV Battery

c) Blue LED to be turned OFF

And when the LV battery voltage drops from 14.1V to 12V, OBEVC shall restart the charging of the LV Battery with blue LED blinking at the rate of 2 sec”

[032] The domain-specific keywords determination module 204 may determine a plurality of domain-specific keywords from the one or more user requirements by prompting a first LLM using a first prompt. In an embodiment, the first prompt may be engineered to prompt the first LLM to output the plurality of domain-specific keywords by determining a set of nouns based on the one or more user requirements corresponding to the test product to be tested. In an embodiment, the set of domain-specific keywords may be but is not limited to nouns related to any device, component, and functionality, etc. corresponding to the product to be tested. In an embodiment, the domain-specific keywords determination module 204 may engineer the first prompt input to the first LLM such that the first LLM may output the plurality of domain-specific keywords in a predefined format. Further, the domain-specific keywords determination module 204 may use a regex checker to list the plurality of domain-specific keywords in a predefined format. In an embodiment, the predefined format may include listing the plurality of domain-specific keywords between square brackets with each keyword separated by a comma. In an embodiment, the first LLM may utilize techniques such as, but not limited to, explainable knowledge ingestion techniques to generate the domain-specific keywords using the domain specific information. In an embodiment, the first prompt may be engineered such that the first LLM may produce results depicting a reasoning behind its decision-making in generating the plurality of domain-specific keywords in accordance with the explainable knowledge ingestion techniques.

[033] According to the exemplary embodiment, the domain-specific keywords determination module 204 may determine the set of domain-specific keywords from the one or more user requirements such as “The vehicle inlet power contacts (SAE J1772) shall be electrically isolated from the battery to avoid electric shock when the connector is removed from the vehicle inlet and detection of the proximity pin disconnection.” as “[vehicle, inlet, power, contacts, SAE J1772, battery, electric shock, connector, detection, proximity pin disconnection].”

[034] The contextual data determination module 206 may sub-include a positional relation determination module 208, and a text extraction module 210. The positional relation determination module 208 may determine a positional relation between each of the plurality of domain-specific keywords with a portion of text data. In an embodiment, the positional relation may be determined based on a lookup of each of the plurality of domain-specific keywords in the text data of the plurality of domain-specific documents. Accordingly, the contextual data determination module 206 may determine the contextual data based on the positional relation between each of the plurality of domain-specific keywords with the text data. Further, the text extraction module 210 may extract the portion of the text data from the plurality of domain-based documents including one or more of the plurality of domain-specific keywords. The contextual data determination module 206 may determine the contextual data based on the extraction of the portion of the text data.

[035] In accordance with the exemplary embodiment, the positional relation determination module 208 may determine the positional relation between each of the plurality of domain-specific keywords with a portion of text data present in the plurality of domain-specific documents. For example, the plurality of domain-specific documents may include a requirement document, an architecture document, and a matrix file. In accordance with the exemplary embodiment, the positional relation may be determined based on the lookup of the domain-specific keyword “SAE J1772” in the text data of each of the plurality of domain-specific documents.

[036] Accordingly, in accordance with the exemplary embodiment, the portion of text data in each of the plurality of domain-based documents in which the domain-specific keyword “SAE J1772” is found or looked-up may include various paragraphs from each of the plurality of domain-based documents.

[037] Accordingly, the text extraction module 210 may extract the plurality of paragraphs as portions of the text data that may include one or more of the plurality of domain-specific keywords. In an embodiment, the portions of the text data extracted by the text extraction module 210 may be in form of a predefined second format including, paragraph, a table, or listed points, etc. Accordingly, in accordance with the exemplary embodiment, the portion of text data extracted from each of the plurality of domain-based documents by the text extraction module 210 that may include the domain-specific keyword “SAE J1772” may include the following exemplary paragraphs:

“Usage of SAE J1772 connector as an interface between AC grid (EVSE) and the OBEVC (On-board EV Charger).”

“Communication and connection between OBVEC Charger and the Grid follow the SAE J1772 connector standard.”

“The SAE J1772 connector majorly consists of 5 pins for handling the operation. Grid Line Voltage, Grid Neutral, Ground, Control Pilot, Proximity Detection.”

“SAE J1772 stands for society of Automotive Engineers standard J1772.”

[038] The knowledge dataset determination module 212 may determine a knowledge dataset by prompting a second LLM using a second prompt and the contextual data. In an embodiment, the knowledge dataset may include a set of questions and a set of answers to each of the set of questions based on the contextual data and the text data of the plurality of domain-based documents. In an embodiment, the second prompt may be engineered to prompt the second LLM to list the set of questions and the set of answers to each of the set of questions based on the contextual data and the text data of the plurality of domain-based documents. In an embodiment, the second LLM may utilize techniques such as, but not limited to, explainable knowledge ingestion techniques to generate the list of the set of questions and the set of answers to each of the set of questions using the domain specific information. In an embodiment, the second prompt may be engineered such that the second LLM may produce results depicting a reasoning behind its decision-making in generating the set of questions and the set of answers to each of the set of questions in accordance with the explainable knowledge ingestion techniques.

[039] The knowledge dataset determination module 212 may determine the knowledge dataset that may include the set of questions and the set of answers to each of the set of questions by prompting the second LLM using the second prompt. In an embodiment, the second prompt

may be engineered to prompt the second LLM to output at least a predefined number of questions and answers to each of the predefined number of questions in detail from each of the plurality of domain-specific documents for each line or paragraphs of the text data extracted by the text extraction module 210.

5 [040] In accordance with the exemplary embodiment, for examples, the second prompt input to the second LLM may include, but is not limited to, “generate at least 2 questions and describe the answer in detail from the document based on the text of each line [*path of contextual data*]”. Accordingly, the knowledge dataset determination module 212 may determine the set of questions and answers to each of the set of questions in detail based on the text data of the
10 plurality of domain specific documents.

[041] According to the exemplary embodiment, the set of questions and answer corresponding to each of the set of questions determined based on the contextual data may include , but is not limited to, the following:

“Q1. What is the purpose of using the SAE J1772 connector in the project?

15 Answer: The SAE J1772 connector is used as an interface between the AC grid (EVSE) and the OBEVC (On-board EV Charger) to provide charge to the HV battery and LV battery and to supply power to auxiliary loads.”

“Q2. What is the purpose of the SAE J1772 connector in the On-board EV Charger system?

20 Answer: The SAE J1772 connector is used as an interface between the AC grid (EVSE) and the OBVEC in the On-board EV Charger system.”

[042] The test cases determination module 214 may determine one or more test cases for testing the at least one feature of the test product by prompting a fourth LLM using a fourth prompt based on the one or more user requirements and the knowledge dataset. In an embodiment, the
25 fourth prompt may be engineered to prompt the fourth LLM to list the one or more test cases corresponding to the one or more user requirements. In an embodiment, the fourth LLM may utilize techniques such as, but not limited to, explainable knowledge ingestion techniques to generate the list of the one or more test cases using the domain specific information. In an embodiment, the fourth prompt may be engineered such that the fourth LLM may produce
30 results depicting a reasoning behind its decision-making in generating the list of one or more test cases in accordance with the explainable knowledge ingestion techniques.

[043] In accordance with the exemplary embodiment, one of the user requirements include:

“When the LV battery is fully charged to 14.1V then, OBEVC shall:

- a) Cut-off the power supply to the LV battery.
- b) Stop charging LV battery.
- 5 c) Blue LED to be turned OFF and when the LV battery voltage drops from 14.1V to 12V, OBEVC shall restart the charging of the LV battery with blue LED blinking at the rate of 2sec.”

[044] Accordingly, the fourth prompt as per the exemplary embodiment may be engineered to include, but is not limited to, as follows:

- 10 “1. Understand the requirement and identify the key components that need to be tested.
2. Identify the scenario that needs to be tested based on the requirements.
3. Create a test case title that clearly describes the scenario being tested and the expected outcome.
4. Review and refine the test case title to ensure that it accurately reflects the requirement and
- 15 the scenario being tested. Test case should start with verify word.”

[045] Further, the test cases determination module 214 may determine the one or more test cases for prompting the fourth LLM using the fourth prompt. According to the exemplary embodiment, one or more test cases determined by the fourth LLM may include, but is not limited to, as follows:

- 20 “1. Verify that OBEVC cuts off the power supply to the LV battery when the LV battery is fully charged to 14.1V.
2. Verify that OBEVC restarts charging LV battery when voltage drops from 14.1V to 12V and blue LED blinks at the rate of 2 sec.”

[046] The test procedures determination module 216 may determine one or more test
25 procedures for the one or more test cases for testing the at least one feature of the test product. The one or more test procedures for the one or more test cases may be determined by prompting a third LLM using a third prompt and the knowledge dataset. In an embodiment, each of the one or more test procedures may include at least one pre-condition, a set action steps to be performed for testing the test product, and at least one expected outcome for the at least one
30 pre-condition.

5 [047] In an embodiment, the third prompt may include the one or more test cases for testing the at least one feature of the test product. In an embodiment, the third prompt may be determined as an output generated by the fourth LLM queried using the fourth prompt. In an embodiment, the third prompt may be engineered to prompt the third LLM to output the one or more test procedures. In an embodiment, the third prompt may be engineered to output the one or more test procedures for the one or more test cases based on the knowledge dataset and the one or more user requirements. In an embodiment, the third LLM may utilize techniques such as, but not limited to, explainable knowledge ingestion techniques to output the one or more test procedures for the one or more test cases using the domain specific information. In 10 an embodiment, the third prompt may be engineered such that the third LLM may produce results depicting a reasoning behind its decision-making in determining the one or more test procedures for the one or more test cases in accordance with the explainable knowledge ingestion techniques.

15 [048] In accordance with the exemplary embodiment, example of the third prompt may include, but is not limited to, as follows:

“Generate the test procedure provide a one-shot example:

Test procedure containing:

1. Precondition/ Test set up.
2. Test Procedure.
- 20 3. Expected result.

Similarly to the example provided generate the result.”

[049] The test procedures determination module 216 may determine the one or more test procedures for the one or more test cases by prompting the third LLM using the third prompt. According to the exemplary embodiment, the third prompt may include the one or more test cases. According to the exemplary embodiment, the one or more test procedure determined by 25 the third LLM based on the third prompt may include, but are not limited to, as follows:

“Pre-condition:

1. Follow steps mentioned in “Flashing Procedure” file.

2. Execute test case Auto_I-00347-01-01_SysQTP_217 to check error/anomaly status before starting this particular test case.
3. Send LV Battery voltage command with 13.5V for charging LV battery to the required voltage. (Message ID: 0*100, Byte0: 0*35, Byte1: 0*4C)
- 5 4. Send LV Battery current command with 20A for charging LV battery to the required current. (Message ID: 0*99, Byte0: 0*01, Byte1: 0*14)

Test Procedure:

1. Connect the LV battery to the OBEVC.
2. Verify LV battery voltage command with 14.1 V for fully charging the LV battery.
10 (Message ID: 0*100, Byte0: 0*32, Byte1:0*85)
3. Monitor the charging process until the LV battery is fully charged to 14.1V.
4. Verify that OBEVC cuts off the power supply to the LV battery when the LV battery is fully charged to 14.1V.
5. Measure voltage at LV O/P port with a multi-meter to ensure that the LV battery voltage
15 is 14.1V.
6. Verify that Blue LED is turned off after the power supply is cut off to the LV battery.

Expected Results:

1. LV battery voltage should be 14.1V (Message ID: 0*100, Byte0: 0*32, Byte1: 0*85).
2. OBVEC should cut off the power supply to the LV battery when the LV battery is fully
20 charged to 14.1V.
3. LV battery is not in charging state. Blue LED should be turned off.”

[050] Accordingly, the test procedure may be used for testing the test product for the pre-condition. The expected output of the test case may be verified by determining success or failure of the expected results of the test procedure for the corresponding test case.

[051] It should be noted that all such aforementioned modules 202-216 may be represented as a single module or a combination of different modules. Further, as will be appreciated by those skilled in the art, each of the modules 202-216 may reside, in whole or in parts, on one device or multiple devices in communication with each other. In some embodiments, each of the
5 modules 202-216 may be implemented as dedicated hardware circuit comprising custom application-specific integrated circuit (ASIC) or gate arrays, off-the-shelf semiconductors such as logic chips, transistors, or other discrete components. Each of the modules 202-216 may also be implemented in a programmable hardware device such as a field programmable gate array (FPGA), programmable array logic, programmable logic device, and so forth. Alternatively,
10 each of the modules 202-216 may be implemented in software for execution by various types of processors (e.g. processor 104). An identified module of executable code may, for instance, include one or more physical or logical blocks of computer instructions, which may, for instance, be organized as an object, procedure, function, or other construct. Nevertheless, the executables of an identified module or component need not be physically located together but
15 may include disparate instructions stored in different locations which, when joined logically together, include the module and achieve the stated purpose of the module. Indeed, a module of executable code could be a single instruction, or many instructions, and may even be distributed over several different code segments, among different applications, and across several memory devices.

[052] As will be appreciated by one skilled in the art, a variety of processes such as, but not limited to, explainable knowledge ingestion, may be employed for engineering prompt input for determining test procedure using various LLMs. For example, the exemplary system 100 and the associated computing device 102 may determine test procedures using LLMs by the processes discussed herein. In particular, as will be appreciated by those of ordinary skill in the
25 art, control logic and/or automated routines for performing the techniques and steps described herein may be implemented by the system 100 and the associated computing device 102 either by hardware, software, or combinations of hardware and software. For example, suitable code may be accessed and executed by the one or more processors 104 on the system 100 to perform some or all of the techniques described herein. Similarly, application specific integrated circuits
30 (ASICs) configured to perform some, or all of the processes described herein may be included in the one or more processors on the system 100.

[053] Referring now to **FIG. 3**, a flow diagram of a method of determining test procedures using large language models, in accordance with an embodiment of present disclosure. In an embodiment, method 300 may include a plurality of steps that may be performed by the processor 104 to determine test procedures.

5 [054] **FIG. 3** is explained in conjunction with **FIGs. 1** and **2**. Each step of the method 300 may be executed by various modules of the computing device 102. At step 302, a plurality of domain-based documents may be received corresponding to a test product to be tested. Further at step 304, one or more user requirements may be determined corresponding to at least one feature of the test product from one of the plurality of domain-based documents. In an
10 embodiment, the plurality of domain-based documents may include a documents including one or more user requirements. Further at step 306, a plurality of domain-specific keywords may be determined from the one or more user requirements by prompting a first LLM using a first prompt. In an embodiment, the first prompt may be engineered to prompt the first LLM to output the plurality of domain-specific keywords by determining a set of nouns based on the
15 one or more user requirements corresponding to the test product to be tested.

[055] Further at step 308, contextual data may be determined by extracting a portion of text data from the plurality of domain-based documents. In an embodiment, the portion of the text data may include one or more of the plurality of domain-specific keywords. Further, in an embodiment, the contextual data may be determined based on determination of a positional
20 relation between each of the plurality of domain-specific keywords with the text data from the plurality of domain-based documents. In an embodiment, the positional relation may be determined based on a lookup of each of the plurality of domain-specific keywords in the text data of the plurality of domain-specific documents. In an embodiment, the contextual data may be determined by extracting the portion of the text data from the plurality of domain-based
25 documents including one or more of the plurality of domain-specific keywords.

[056] Further at step 310, a knowledge dataset may be determined by prompting a second LLM using a second prompt and the contextual data. In an embodiment, the knowledge dataset may include a set of questions and a set of answers to each of the set of questions based on the contextual data and the text data of the plurality of domain-based documents. In an
30 embodiment, the second prompt may be engineered to prompt the second LLM to list the set of questions and the set of answers to each of the set of questions based on the contextual data and the text data of the plurality of domain-based documents.

[057] Further at step 312, one or more test cases may be determined for testing the at least one feature of the test product by prompting a fourth LLM using a fourth prompt. In an embodiment, the fourth prompt may be engineered to prompt the fourth LLM to list the one or more test cases corresponding to the one or more user requirements.

5 **[058]** Further at step 314, one or more test procedures may be determined for the one or more test cases for testing the at least one feature of the test product by prompting a third LLM using a third prompt and the knowledge dataset. In an embodiment, each of the one or more test procedures may include at least one pre-condition, a set action steps to be performed for testing the test product, and at least one expected outcome for the at least one pre-condition. In an
10 embodiment, the third prompt may include the one or more test cases for testing the at least one feature of the test product. In an embodiment, the third prompt may be determined as an output generated by the fourth LLM queried using the fourth prompt. In an embodiment, the third prompt may be engineered to output the one or more test procedures for the one or more test cases based on the knowledge dataset and the one or more user requirements.

15 **[059]** In an embodiment, challenges for determining test procedures using LLMs for generating consistent codes may include but are not limited to the primary challenge is providing an LLM with domain knowledge to feed it with external or domain-specific knowledge. Conventionally, this was done by using Retrieval Augmented Generation (RAG), , or by fine-tuning the entire model which has its own set of difficulties. Large-scale applications may find
20 fine-tuning problematic due to the substantial computing challenges involved. Conversely, using RAG necessitates sending structural data to the LLMs in an efficient manner so that they can understand what is being taught. But in real-world situations, the data is frequently large and disorganized. As a result, chunking techniques-which split documents into smaller chunks for prediction-are mostly used by RAG. However, this method presents a risk of factual loss
25 and inaccuracy, which could result in inaccurate predictions.

[060] Thus, the disclosed method and system tries to overcome the technical problem of determining test procedures through a method and system of determining test procedures using Large Language Models against such challenges. In an embodiment, advantages of the disclosed method and system may include but is not limited to the substantial reduction in
30 hallucination, which refers to the generation of inaccurate or fictional information by the model. In determining test procedures using Large Language Models (LLMs), the proposed method and system ensures that the generated test cases are accurate, reliable, and aligned with the actual requirements of the domain. Furthermore, the approach focuses on extracting and

condensing only the relevant knowledge needed for a specific query. This targeted extraction of pertinent information helps streamline the data input to the LLM, eliminating unnecessary details and noise. By condensing the vast knowledge data to include only what is relevant, the proposed method enhances the model's efficiency and effectiveness in generating precise test procedures. Additionally, the approach tackles the challenge of dealing with voluminous and unstructured real-world data.

[061] As will be appreciated by those skilled in the art, the techniques described in the various embodiments discussed above are not routine, or conventional, or well-understood in the art. The techniques discussed above provide for determining test procedures using Large Language Models.

[062] In light of the above-mentioned advantages and the technical advancements provided by the disclosed method and system, the claimed steps as discussed above are not routine, conventional, or well understood in the art, as the claimed steps enable the following solutions to the existing problems in conventional technologies. Further, the claimed steps bring an improvement in the functioning of the device itself as the claimed steps provide a technical solution to a technical problem.

[063] The specification has described method and system for determining test procedures using Large Language Models (LLMs). The illustrated steps are set out to explain the exemplary embodiments shown, and it should be anticipated that ongoing technological development will change the manner in which particular functions are performed. These examples are presented herein for purpose of illustration, and not limitation. Further, the boundaries of the functional building blocks have been arbitrarily defined herein for the convenience of the description. Alternative boundaries can be defined so long as the specified functions and relationships thereof are appropriately performed. Alternatives (including equivalents, extensions, variations, deviations, etc., of those described herein) will be apparent to persons skilled in the relevant art(s) based on the teachings contained herein. Such alternatives fall within the scope and spirit of the disclosed embodiments.

[064] It is intended that the disclosure and examples be considered as exemplary only, with a true scope of disclosed embodiments being indicated by the following claims.

We Claim:

1. A method (300) of determining test procedures using large language models (LLMs), the method comprising:

receiving (302), by a processor (104), a plurality of domain-based documents corresponding to a test product to be tested;

determining (304), by the processor (104), one or more user requirements corresponding to at least one feature of the test product from one of the plurality of domain-based documents;

determining (306), by the processor (104), a plurality of domain-specific keywords from the one or more user requirements by prompting a first LLM based on a first prompt;

determining (308), by the processor (104), contextual data by extracting a portion of text data from the plurality of domain-based documents,

wherein the portion of text data comprises one or more of the plurality of domain-specific keywords;

determining (310), by the processor (104), a knowledge dataset by prompting a second LLM based on a second prompt and the contextual data,

wherein the knowledge dataset comprises a set of questions and a set of answers to each of the set of questions based on the contextual data and the text data of the plurality of domain-based documents, and

wherein the second prompt is engineered to prompt the second LLM to list the set of questions and the set of answers to each of the set of questions based on the contextual data and the text data of the plurality of domain-based documents; and

determining (314), by the processor (104), one or more test procedures for one or more test cases for testing the at least one feature of the test product by prompting a third LLM based on a third prompt and the knowledge dataset,

wherein the third prompt comprises the one or more test cases for testing the at least one feature of the test product.

2. The method (300) as claimed in claim 1, wherein the third prompt is determined as an output generated by a fourth LLM queried based on a fourth prompt,

wherein the third prompt is engineered to output the one or more test procedures for the one or more test cases based on the knowledge dataset and the one or more user requirements, and

wherein the third prompt is engineered to prompt the third LLM to output the one or more test procedures.

3. The method (300) as claimed in claim 2, wherein each of the one or more test procedures comprises at least one pre-condition, a set action steps to be performed for testing the test product, and at least one expected outcome for the at least one pre-condition.

4. The method (300) as claimed in claim 1, wherein the contextual data is determined based on determination of a positional relation between each of the plurality of domain-specific keywords with the text data,

wherein the positional relation is determined based on a lookup of each of the plurality of domain-specific keywords in the text data of the plurality of domain-based documents.

5. The method (300) as claimed in claim 1, wherein the first prompt is engineered to prompt the first LLM to output the plurality of domain-specific keywords by determining a set of nouns based on the one or more user requirements corresponding to the test product to be tested.

6. The method (300) as claimed in claim 1, wherein the fourth prompt is engineered to prompt the fourth LLM to list the one or more test cases corresponding to the one or more user requirements.

7. A system (100) of determining test procedures using large language models (LLMs), comprising:

a processor (104); and

a memory (106) communicably coupled to the processor (104), wherein the memory (106) stores processor-executable instructions, which, on execution, cause the processor (104) to:

receive a plurality of domain-based documents corresponding to a test product to be tested;

determine one or more user requirements corresponding to at least one feature of the test product from one of the plurality of domain-based documents;

determine a plurality of domain-specific keywords from the one or more user requirements by prompting a first LLM based on a first prompt;

determine a contextual data by extracting a portion of a text data from the plurality of domain-based documents,

wherein the portion of the text data comprises one or more of the plurality of domain-specific keywords;

determine a knowledge dataset by prompting a second LLM based on a second prompt and the contextual data,

wherein the knowledge dataset comprises a set of questions and a set of answers to each of the set of questions based on the contextual data and the text data of the plurality of domain-based documents, and

wherein the second prompt is engineered to prompt the second LLM to list the set of questions and the set of answers to each of the set of questions based on the contextual data and the text data of the plurality of domain-based documents; and

determine one or more test procedures for one or more test cases for testing the at least one feature of the test product by prompting a third LLM based on a third prompt and the knowledge dataset,

wherein the third prompt comprises the one or more test cases for testing the at least one feature of the test product.

8. The system (100) as claimed in claim 7, wherein the third prompt is determined as an output generated by a fourth LLM queried based on a fourth prompt,

wherein the fourth prompt is engineered to output the one or more test procedures for the one or more test cases based on the knowledge dataset and the one or more user requirements, and

wherein the third prompt is engineered to prompt the third LLM to output the one or more test procedures.

9. The system (100) as claimed in claim 8, wherein each of the one or more test procedures comprises at least one pre-condition, a set action steps to be performed for testing the test product, and at least one expected outcome for the at least one pre-condition.

10. The system (100) as claimed in claim 7, wherein the contextual data is determined based on determination of a positional relation between each of each of the plurality of domain-specific keywords with the text data,

wherein the positional relation is determined based on a lookup of each of the plurality of domain-specific keywords in the text data of the plurality of domain-based documents.

11. The system (100) as claimed in claim 7, wherein the first prompt is engineered to prompt the first LLM to output the plurality of domain-specific keywords by determining a set of nouns based on the one or more user requirements corresponding to the test product to be tested.

12. The system (100) as claimed in claim 7, wherein the fourth prompt is engineered to prompt the fourth LLM to list the one or more test cases corresponding to the one or more user requirements.

Dated this 12th day of April 2024

--Digitally Signed--
Bhanu Prasad (INPA No: 3253)
Head, IPR Dept.,
L&T Technology Services Limited,
DLF 3rd Block, 2nd Floor,
Manapakkam, Chennai, TN, 600089.

ABSTRACT

METHOD AND SYSTEM OF DETERMINING TEST PROCEDURES USING LARGE LANGUAGE MODELS

A method (300) and system (100) of determining test procedures using large language models (LLMs) is disclosed. A processor (104) receives a plurality of domain-based documents corresponding to a test product to be tested. One or more user requirements are determined corresponding to at least one feature of the test product from one of the plurality of domain-based documents. A plurality of domain-specific keywords is determined from the one or more user requirements. A contextual data is determined by extracting a portion of a text data from the plurality of domain-based documents. A knowledge dataset is determined by prompting a second LLM based on a second prompt and the contextual data. One or more test procedures for one or more test cases for testing the at least one feature of the test product by prompting a third LLM based on a third prompt and the knowledge dataset.

[To be published with FIG. 1]

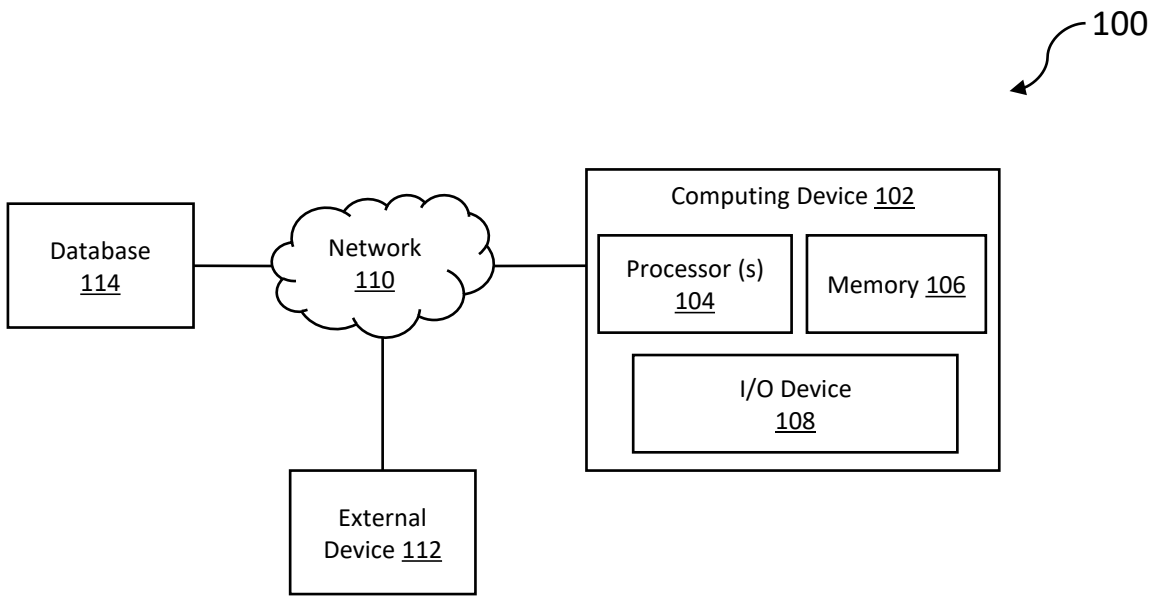


FIG. 1

--Digitally Signed--
Bhanu Prasad (INPA No: 3253)
Head, IPR Dept.,
L&T Technology Services Limited,
DLF 3rd Block, 2nd Floor,
Manapakkam, TN, Chennai - 600089.

200

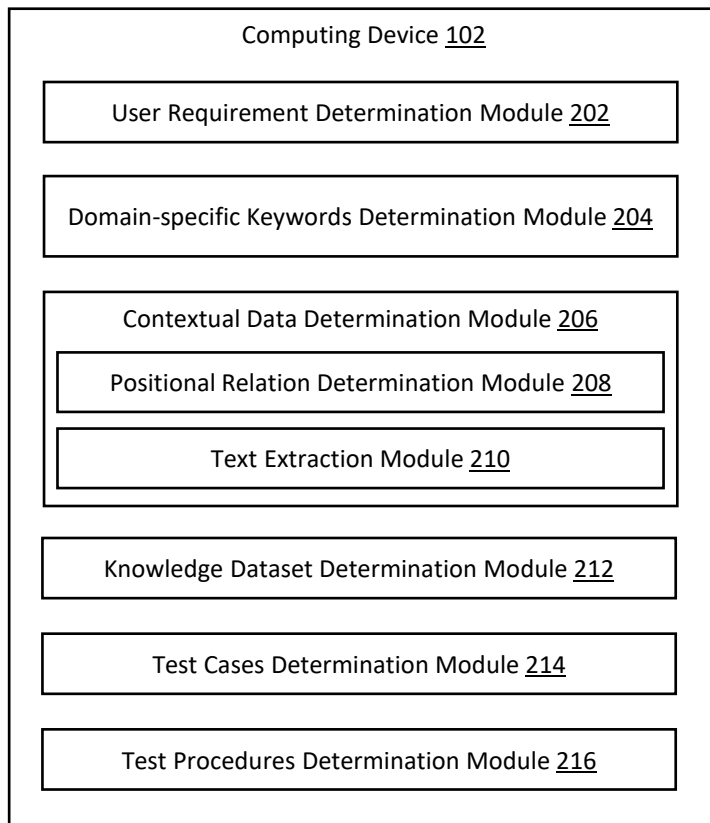
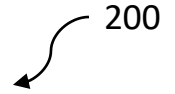


FIG. 2

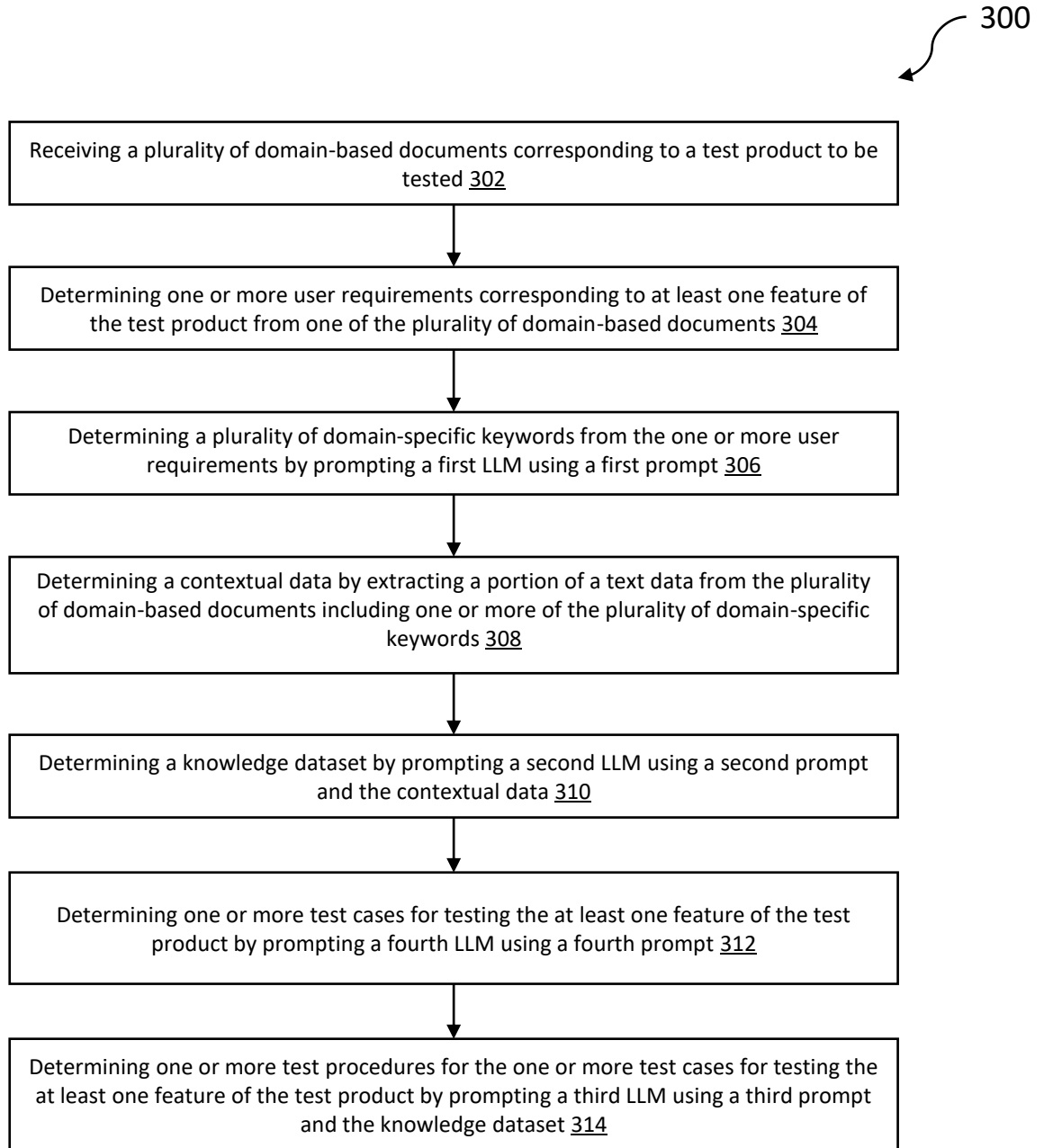


FIG. 3

--Digitally Signed--
Bhanu Prasad (INPA No: 3253)
Head, IPR Dept.,
L&T Technology Services Limited,
DLF 3rd Block, 2nd Floor,
Manapakkam, TN, Chennai - 600089.