IIT MADRAS
Indian Institute of Technology Madras

Technology Transfer Office
TTO – IPM Cell

Industrial Consultancy & Sponsored Research (IC&SR)

A SYSTEM AND METHOD FOR BUG IDENTIFICATION AND FAULT LOCALIZATION THROUGH DOMAIN-ONTOLOGY

IITM Technology Available for Licensing

Problem Statement

- Debugging is a major component that ensures proper functioning of the program. The **existing debuggers and fault localization systems are restricted only to programs** and do not consider any knowledge apart from the programs.
- Applications are usually developed in a programming language, and are to be operated in a particular domain. Typically, debuggers find it difficult to efficiently consider the corresponding domain-knowledge. Additionally, standardized procedures for instrumenting programs and generating execution-traces are challenges faced while performing a validation task.
- Further, the thoughts of a programmer, and the reason and explanations for the code written by the developers are not documented in existing debugging methods. There are separate formal representations available to the programs, execution of the programs and the domain-knowledge. However, these **representations are not inter-operative with each other**.
- Thus, there is need for a system and method that automatically identifying bugs and generating explanations of the identified bugs.

Technology Category/ Market

Category - Information & Communication Technology (ICT), Software Debugging.
Applications - Software engineering, software validation, software testing, software debugging.

Market - Debugging software market size was valued at around USD 638 Million in 2021 and is projected to reach USD 2022 Million by 2030, growing at a **CAGR of 13.66%** from 2023 to 2030.

Intellectual Property

- IITM IDF Ref. **2137**
- **IN 202141027824**
- **PCT/IN2022/050562**

TRL (Technology Readiness Level)

TRL - 4, Technology validated in lab.

Technology

The present invention provides a method for automatically identifying bugs and a portion of a code responsible for bugs in a program.

Method

1 • The method integrates domain-trace and domain-ontology to identify bugs. A **domain-ontology-based Program Assertion (DOPA) framework (Fig.1)** is configured to integrate formal representations available to the program, execution of the program, and domain-knowledge, thereby making these representations inter-operative to identify bugs.

2 • The DOPA framework instruments a program provided by a user, wherein **execution-trace file** from the instrumented program is computed and stored during run-time using a **Resource Description Framework (RDF). Fig. 2.**

3 • The **domain-ontology (Fig.4)** is created by the user and subsequently, a domain-trace is generated. Further, the domain-trace and the domain-ontology are integrated and fed inside a reasoner to identify the bugs in the program.

4 • Subsequently, the bugs and individuals of the bugs are identified. Thereafter, explanations of the **identified bugs are generated using items of execution-trace and domain-ontology. (Refer Fig. 3)**

5 • Subsequently, the identified bugs and the identified explanations related to the bugs are presented to the user**. (refer Fig.5)**

Key Features / Value Proposition

1. **Automatic debugging using machine-processable domain-ontology**, wherein portions of the code responsible for the bugs are automatically identified.
2. Ontology is developed as a **one-time job and can be used for all the programs** developed for a domain.
3. The invention works with the programs developed in **any imperative languages** and the ontology useful in documentation plays a key role to identify bugs in the programs.
4. Can be used in any applications developed in **Java, C#, Scala, or other programming languages**.

Research Lab

Prof. Rupesh Nasre & Prof. P. Sreenivasa Kumar,
Dept. of Computer Science & Engg., IIT Madras

CONTACT US
Dr. Dara Ajay, Head
Technology Transfer Office,
IPM Cell- IC&SR, IIT Madras

IITM TTO Website:
https://ipm.icsr.in/ipm/

Email: smipm-icsr@icsrpis.iitm.ac.in
sm-marketing@imail.iitm.ac.in
Phone: +91-44-2257 9756/ 9719

Fig. 1. illustrates a system depicting the components of a system for bug identification and fault localization through domain-ontology.

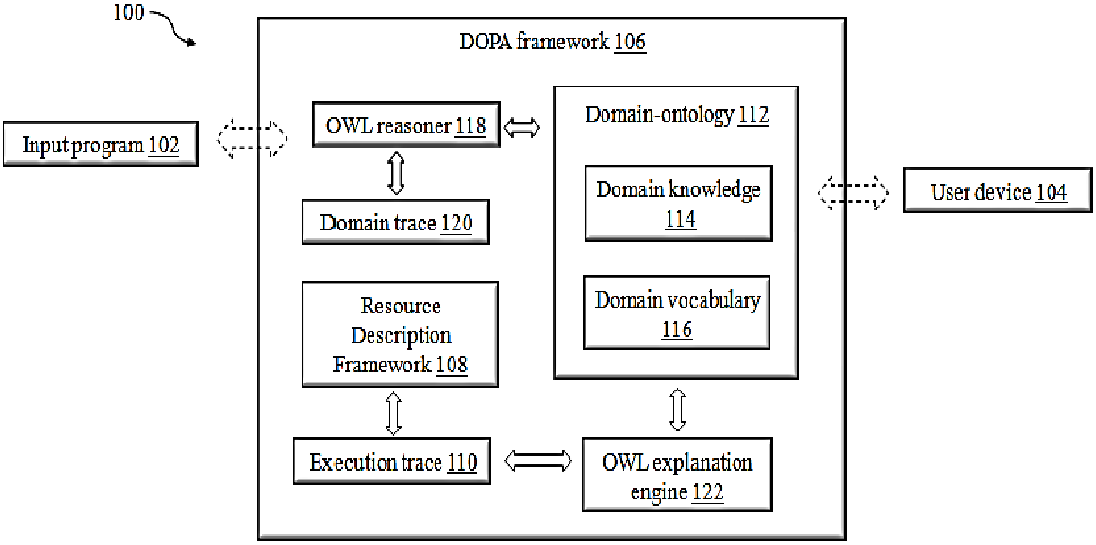


Fig. 2. illustrates a method for instrumenting a program by a framework.

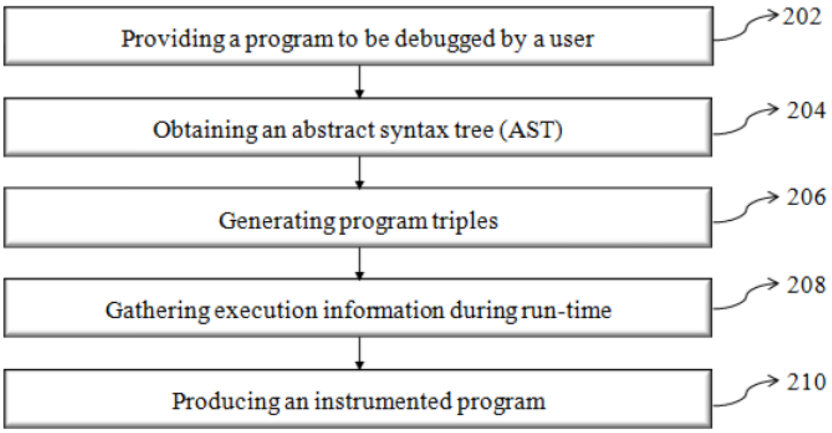

Providing a program to be debugged by a user
202
Obtaining an abstract syntax tree (AST)
204
Generating program triples
206
Gathering execution information during run-time
208
Producing an instrumented program
210

Fig. 3. illustrates a method for computing an execution trace-file from an instrumented program.

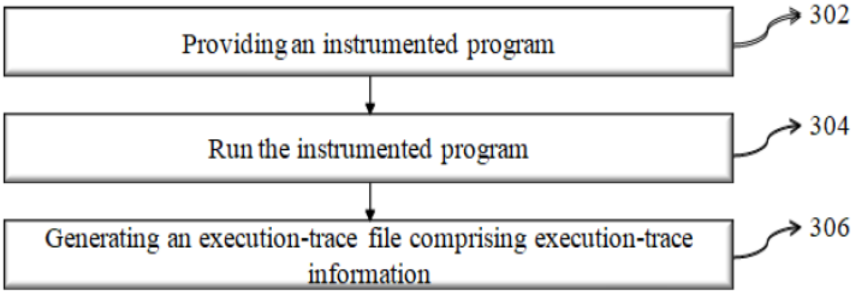

Providing an instrumented program
302
Run the instrumented program
304
Generating an execution-trace file comprising execution-trace information
306

Fig. 4. illustrates a method for creating domain-ontology by a user.

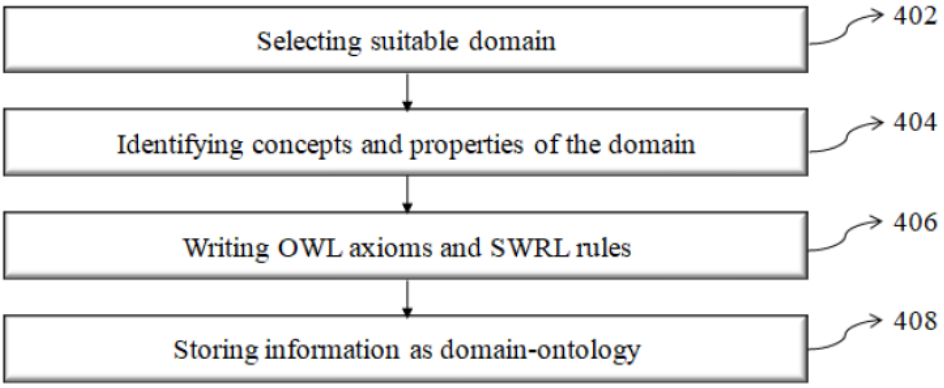
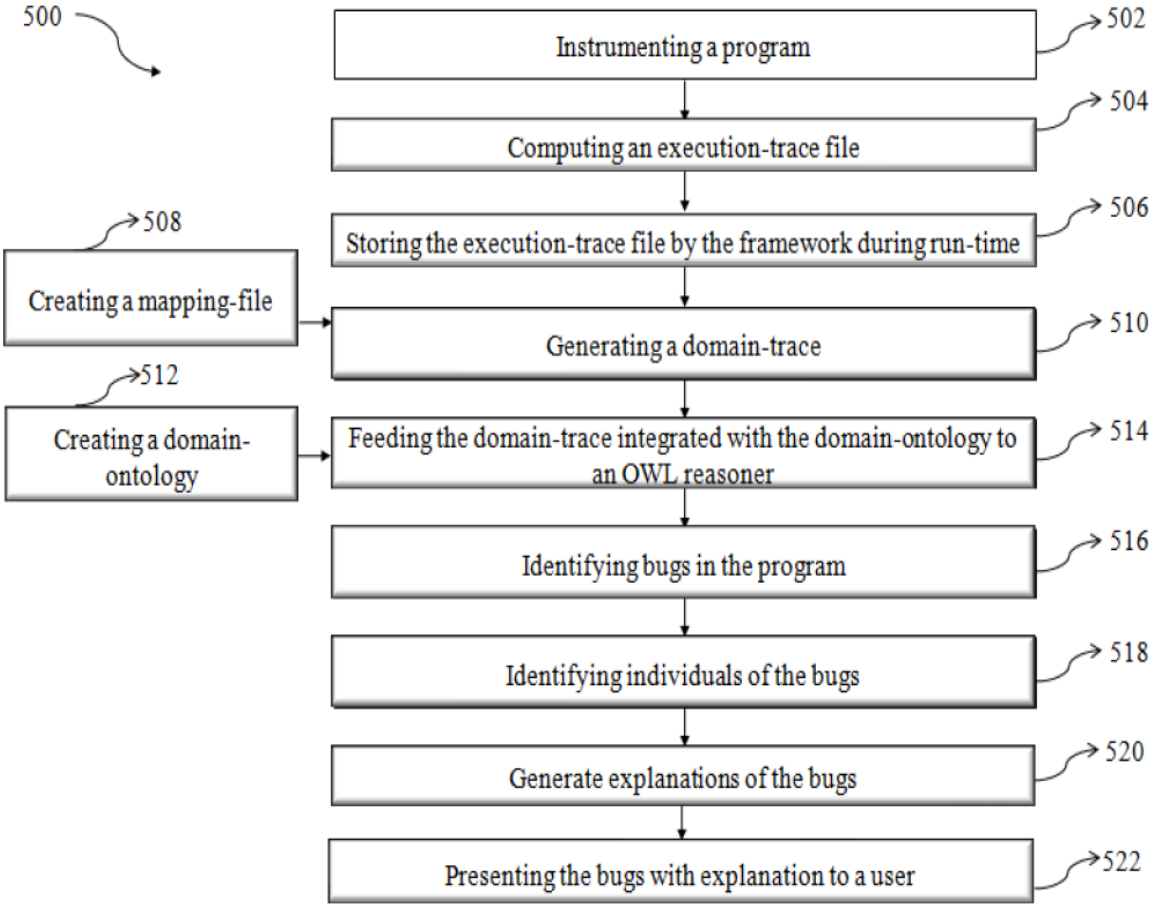

Selecting suitable domain
402
Identifying concepts and properties of the domain
404
Writing OWL axioms and SWRL rules
406
Storing information as domain-ontology
408

Fig. 5. illustrates a method for identifying bugs and presenting the identified bugs to a user by using domain-trace and domain-ontology.


500
Instrumenting a program
502
Computing an execution-trace file
504
Storing the execution-trace file by the framework during run-time
506
508
Creating a mapping-file
Generating a domain-trace
510
512
Creating a domain-ontology
Feeding the domain-trace integrated with the domain-ontology to an OWL reasoner
514
Identifying bugs in the program
516
Identifying individuals of the bugs
518
Generate explanations of the bugs
520
Presenting the bugs with explanation to a user
522