TEST Engineering & Management

July - August 2020
ISSN: 0193-4120 Page No. 5932 - 5937

Research on Computer Fault Diagnosis Based on Ontology and Rules

Yuhua Peng[1], Ying Wang[2,*]

[1]School of Artificial Intelligence, Wuchang University of Technology, Wuhan, Hubei, China, 430223
[2]State Key Laboratory of Materials Processing and Die & Mold Technology, Huazhong University of Science and Technology, Wuhan, Hubei, China, 430074

Article Info
Volume 83
Page Number: 5932 - 5937
Publication Issue:
July - August 2020

Article History
Article Received: *25 April 2020*
Revised: *29 May 2020*
Accepted: *20 June 2020*
Publication: *28 August 2020*

Abstract

The technology in the research of computer fault diagnosis based on ontology and rules effectively solves the computer fault diagnosis by applying intermittent strategy aperture. Conventional computer diagnosis solutions can not be effectively solved, and the engine reasoning model is derived. The successful development of computer fault diagnosis research based on ontology and rules will greatly improve the work efficiency of computer maintenance personnel to ensure normal work operations.

Keywords: *Computer, Fault Diagnosis, Rule-based Reasoning, SWRL, Ontology;*

1. **Introduction**

Ontology is a term derived for computer science in the history of computational law. It is a regularized conceptual model. However, in the wide range of practical computer applications, it also bears the inconvenience caused by various faults[1-3]. Diagnosing and quickly determining the faults is an urgent need to bring a good environment for work and life. It detects computer failures based on the body and rules, accurately determines the cause of the failure through different rules, and records the theoretical cause of the failure with information, which is more efficient and accurate than previous diagnosis. The most important thing is that the ontology and rule detection can be used with various equipment, and it has the development status and practical convenience[4-6].

1. Computer fault diagnosis system framework based on ontology and rules

The system studied in the article is used to diagnose computer faults. It can quickly determine the cause of the fault and provide users with troubleshooting methods based on the computer's fault phenomenon and through ontology and rule-based reasoning. The system framework is shown in Figure 1.

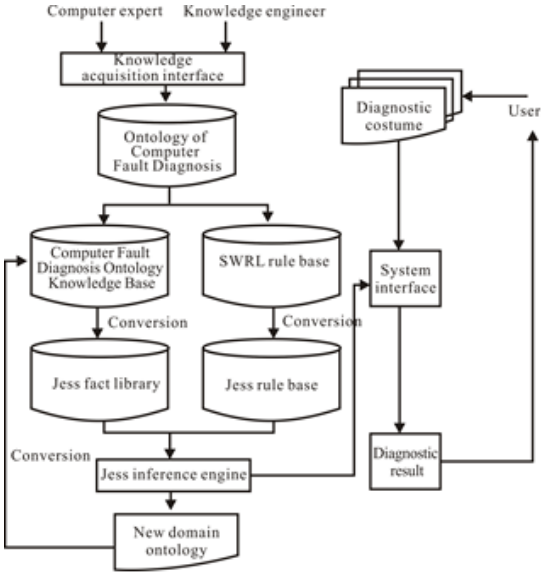

Figure 1 Frame diagram of computer fault diagnosis system.

1) Ontology in the field of computer fault diagnosis is the conceptual basis of the entire system, provides

Published by: The Mattingley Publishing Co., Inc.

July - August 2020
ISSN: 0193-4120 Page No. 5932 - 5937

a hierarchical structure of the ontology, and is also the basis for building rules.

2) Computer fault diagnosis ontology knowledge base is used to support the standard framework of ontology. OWLDL is used to construct ontology. The main constituent elements of ontology are concepts, attributes and examples. It also contains the hidden information of the domain ontology after description logic reasoning.

3) Use the ontology reasoning engine based on description logic-RACER to reason about ontology, thereby checking ontology consistency, discovering hidden information, establishing better hierarchical relationships, and making the information in the OWL knowledge base complete.

4) SWRL is a rule language based on ontology. Combining the concepts, attributes and examples of ontology, the SWRL rule base is constructed.

5) Invoke the SWRLJESSTAB plug-in to convert the OWL knowledge base into a Jess fact base, and convert the SWRL rule base into a Jess rule base.

6) Use the Jess inference engine, combined with the fact base and the rule base, to make inferences and get new facts.

7) Update and expand the ontology knowledge base. Call the SWRLJESSTAB plug-in to format and save the results inferred by Jess.

2. Computer fault diagnosis ontology modeling

The purpose of this system is to help people quickly determine the cause of the failure, find a troubleshooting method, and ensure the smooth progress of various tasks. The object of ontology research is computer fault diagnosis knowledge. The target users of the computer fault diagnosis body are: computer users and computer maintenance personnel. The data source of the system is mainly computer maintenance books, and some supplementary information is collected from the Internet. Due to the complexity of computer composition and diagnosis and the limitation of the length of this article, the ontology constructed in this article mainly starts with computer hardware.

2.1. Definition of ontology concept

When the computer is running, it is often unable to run due to some hardware or software failures, which seriously affects the normal use of the computer. Computer hardware failure refers to the failure of the board card in the computer and the failure of external equipment and other hardware caused by poor contact, performance degradation, damage to circuit components or mechanical problems. The classes of ontology are usually also called concepts. Computer hardware refers to the physical components of the computer. When defining the concept of computer ontology, starting from the composition of its main components, a top-down approach is used to establish a conceptual hierarchy. The hardware system usually consists of CPU (central processing unit), memory (including memory, hard disk, etc.), input devices (keyboard, mouse, etc.), output devices (display, printer, speakers, etc.), interface devices (motherboard, graphics card, network card, sound card , CD-ROM) and other components. In view of the above characteristics, the main body of this research includes: computer hardware, diagnostic and maintenance tools and status. Computer hardware is divided into six categories: CPU and its cooling fan, memory, input device, output device, interface device, and power supply. The specific classification is shown in Figure 2.

Published by: The Mattingley Publishing Co., Inc.

July - August 2020
ISSN: 0193-4120 Page No. 5932 - 5937

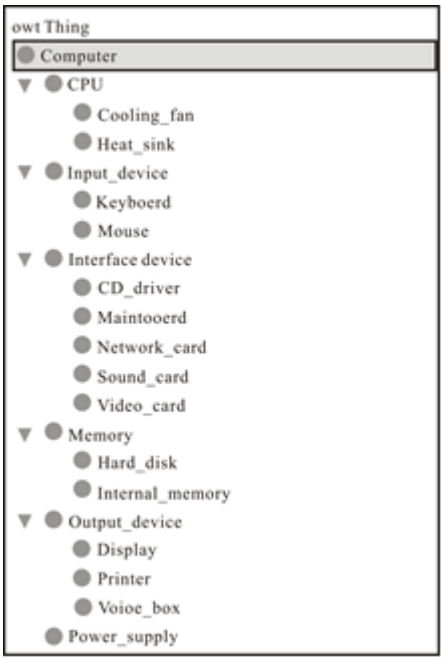

Figure 2 Classification hierarchy of computer hardware categories.

2.2. Establishment of ontology attributes

Computer hardware failure usually results in failure to boot, system failure, failure of a certain device to operate normally, crashes, blue screen and other failure phenomena. In severe cases, it is often accompanied by hot, buzzing, and electric sparks. Therefore, the attribute of the computer body is mainly to describe the above various failure phenomena that the computer may involve, such as: crash, blue screen, hot, buzzing, etc. The attribute relationship of the body part of a diagnostic rule is defined in the form of a predicate of A (x, y), where A represents a predicate of a fault phenomenon, and x and y are variables, OWL instances or data values. For example: isSilent (x, y) means silent phenomenon; means noise phenomenon.

The attributes of the head part of the diagnostic rule are defined in the form of C (x, y), where C represents a predicate for troubleshooting actions or operations, x and y are also variables, OWL instances or data values. For example, Replace (x, y) represents the replacement operation attribute, Clean (x, y) represents the cleaning operation attribute, and Adjust (x, y) represents the adjustment operation attribute. When creating an attribute, you must also limit the scope of the domain and value range.

Instances are primitives used to describe specific semantic concepts in the ontology. After establishing the concept and attributes of the ontology, continue to complete the creation of the instance. The process of creating an instance is: first determine a class, then enter the instance name, and finally select or fill in the value of the relevant attribute.

3. Construction of computer fault diagnosis rules based on SWRL

3.1. SWRL rule construction

In the Semantic Web, SWRL rules are used to support rule system interoperability. SWRL allows users to write Horn-like rules, based on the ontology, using ontology instances and attributes to build Atom clauses. These clauses then form the head and body in the Imp, and finally form the inference formula. Rules can be used to infer new knowledge from the existing OWL knowledge base.

SWRL rules reason about OWL examples, mainly based on OWL classes and attributes. For example, a SWRL rule expresses: a person with a male sibling has a brother, requiring the concepts ‘person’, ‘male’, ‘sibling’ and ‘brother’ to be captured in OWL. Intuitively, the concept of person and male can be recorded by an OWL class. This OWL class is called Person and has a subclass Man; the relationship between sibling and brother can be expressed by the OWL attributes hasSibling and hasBrother, which are attached to Person. The rules in SWRL are:

Person (?x1) hasSibling(?x1,?x2) Man(?x2)→hasBrother(?x1,?x2)

The above relationship cannot be realized in OWL. It can be seen that the combination of ontology and rules overcomes the defects of OWLDL in reasoning, and provides more powerful knowledge representation and reasoning capabilities.

3.2. Construction of computer fault diagnosis rules

On the basis of the established computer fault diagnosis ontology, the diagnosis rules are constructed. The following describes the process of

Published by: The Mattingley Publishing Co., Inc.

July - August 2020
ISSN: 0193-4120 Page No. 5932 - 5937

establishing specific computer fault diagnosis rules. To extract knowledge from the book "Computer Software and Hardware Maintenance From Entry to Proficiency", there are two main reasons for the phenomenon of high CPU temperature, computer crash, or black screen in the CPU heat dissipation fault: (1) CPU cooling fan The problem can be solved by replacing the cooling fan or reinstalling the cooling fan; (2) CPU heat sink problem, the heat sink and the CPU are in poor contact, you can reinstall the CPU heat sink and apply silica gel on the heat sink.

1) CPU cooling fan problem, replace the cooling fan solution:

Computer is x, CPU is y, high state is used to describe the temperature is too high. The isCrash attribute indicates that the computer crashes, the hasTemperature attribute indicates that the CPU temperature is too high, the hasPart attribute connects y and z, indicating that y has a cooling fan z, and finally the z abnormal phenomenon represented by the isAbnormal attribute is introduced. The troubleshooting method is represented by the Replace attribute, and replace the cooling fan z with a screwdriver. Among them, the screwdriver is directly represented by the instance Screwdriver_1 as a parameter of the attribute Replace, which can improve the efficiency of reasoning.

These three rules describe the second reason. Check whether the heat sink is in good contact with the CPU. If the contact is not good, reinstall the CPU heat sink and coat the heat sink with silica gel. The isPoorContacted attribute is used to indicate poor contact, here describes the poor contact between the CPU and the heat sink. Coat attribute represents the operation coating, here is the description of applying silica gel on the heat sink. The two parameters Heat_sink_1 (heat sink) and Silica_gel_1 (silica gel) in the Coat attribute are written in the rules in the form of examples.

Generally, the fault reasoning process based on ontology and rules is shown in Figure 3.

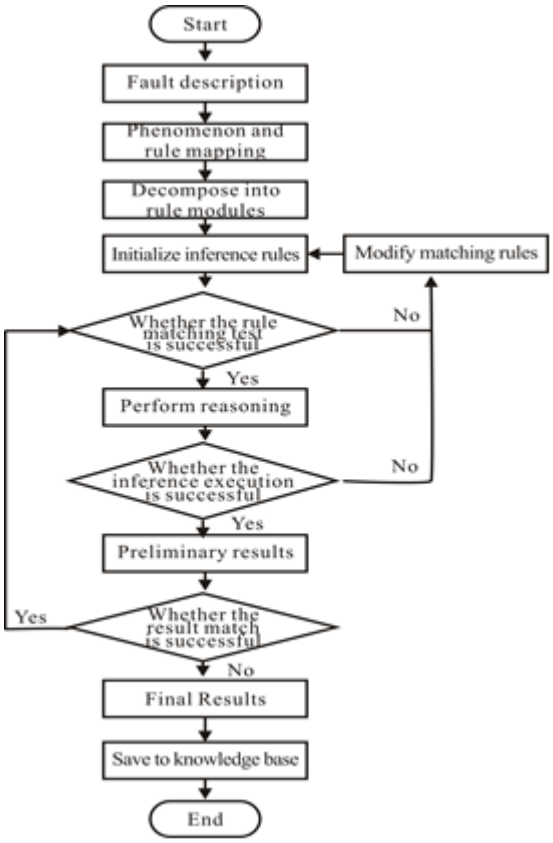

Figure 3. Fault diagnosis reasoning block diagram.

The JESS inference engine used in the fault diagnosis system is the core component of the entire system application. Its main function is to coordinate and control the operation of the system, and use the relevant knowledge in the knowledge base and the specific rule execution in the rule library through the built-in reasoning algorithm. Reasoning process. The efficiency of the inference engine largely depends on the matching algorithm of the inference engine. The JESS inference engine provides efficient forward and backward inference by implementing the Rete matching algorithm, using the characteristics of time redundancy and structural similarity in the diagnosis system. Effectively reduce the number of matching operations, so the diagnosis system adopts JESS reasoning platform, which has the characteristics of less time-consuming and fast response.

4. Realization of computer failure reasoning based on SWRL

4.1. Inference Engine Jess

A large number of rule engines work in Java, and

Published by: The Mattingley Publishing Co., Inc.

July - August 2020
ISSN: 0193-4120 Page No. 5932 - 5937

many are open source software available. There is currently no reasoning engine specifically for SWRL. Commonly used reasoning engines include JESS, Prolog, CLIPS, etc. We chose Jess as the reasoning engine of SWRL because it works seamlessly with Java, has a wide user base, has many documents, is easy to use and configure, and Jess is small, flexible, compatible with CLIPS, and is the fastest known rule engine of. The Jess system consists of a rule base, a fact base and an inference engine. The inference engine matches the facts in the fact base with the rules in the rule base. These rules can assert new facts and put them in the fact base or perform Java functions.

4.2. Computer failure reasoning process

The computer ontology and rules in this article are constructed on the basis of ontology creation software Protégé 3.4.7 and plug-in SWRLEditor. SWRL Editor integrates JESS rule engine. Once the relevant OWL concepts and SWRL rules are represented in Jess, the Jess rule engine can perform inference. As the rule is activated, new Jess facts are inserted into the fact base. These facts will be used in future reasoning. When the reasoning process is over, these facts are transformed into OWL knowledge.

The interaction between the SWRL editor and the Jess rule engine is user-driven. OWL knowledge and SWRL rules are transformed into Jess, using these knowledge and rules to perform reasoning, and the resulting Jess facts are transformed back into Protégé-OWL in the form of OWL knowledge, all under the control of the user.

On the basis of the established ontology, the specific reasoning elements involved in the fault are determined by extracting the ontology concepts, and the reasoning rules are constructed according to the elements. The inference result of the inference rule will be filled with the fault attribute value of the native component. And the attribute value can be rewritten into the ontology file, and the reasoning result can be saved to the ontology to achieve the storage and sharing of knowledge.

In order to illustrate the process of establishing specific rules, the following first describes the failure phenomenon of "excessive noise" when the machine is running. According to the knowledge extraction in the technical documentation of the machine, one reason for the excessive noise during the operation of the machine is that the working chain is too loose, then establish a rule: Chain(?z)∧hasNoise(?x,Big)∧hasPart(?x, ?z)→isLoose(?x,?z)(1).

In rule (1), the working chain is denoted as "z", the parts of the machine are denoted as "x", and "Big" is used to describe a relatively large degree of noise, that is, abnormal noise. "HasPart" is used to limit the working chain "z" under the part "x" of the machine (it is possible that other parts of the machine also have working chains). "IsLoose" connects "x" and "z" as an attribute relationship, which means that the working chain of the native component "x" is too loose. Among them, "x" and "z" are variables, which are generally an instance of the class.

The establishment of the above rules basically reflects the establishment process of the local fault rules. In the process of establishing rules, I hope to achieve the following indicators: According to the method established by the above rules, other fault rules are also established according to this method. Table 1 shows the details of some rules.

Table 1. Machine fault diagnosis rules (partial)

Category	SWRL rules
Signal lamp	Unlighted(?y,?x)∧Fuse(?f)∧hasPart(?x,?f)→BurntOut(?y,?f)

Published by: The Mattingley Publishing Co., Inc.

July - August 2020
ISSN: 0193-4120 Page No. 5932 - 5937

failure	Unlighted(?y,?x)∧BurntOut(?y,?f)→Replace(?y,?f)
	Unlighted(?x,?y)→isDamaged(?x,?y)
	Unlighted(?x,?y)∧isDamaged(?x,?y)→Replace(?x,?y)
	isSilent(Driver-Device-1,?x)→isDamaged(Driver-Device-1,?x)
Bell failure	isSilent(Driver-Device-1,?x)∧isDamaged(Driver-Device-1,?x)→Replace(Drive r-Device-1,?x)
	Chain(?z)∧hasNoise(?x,Big)∧hasPart(?x,?z)→isLoose(?x,?z)
Running	hasNoise(?x,Big)∧isLoose(?x,?z)→Tension(?z,Aajustable-wrench-1)
noise	Chain(?z)∧hasNoise(?x,Big)→hasDryfriction(?z,Chain-Guide-Rail-1)
failure	hasNoise(Planetary-Reduction-Box-1,Big)→isDamaged(Planetary-Reduction-Box-1,Oil-Sealing-1)
…	…

5. Conclusion

Equipment fault diagnosis is a very complicated project, especially for military equipment with higher equipment requirements. In the native fault diagnosis system studied in this paper, the fault diagnosis rules adopt the OWL ontology form, which is convenient for storing and sharing knowledge. The architecture based on SWRL rules and JESS reasoning engine has the advantages of strong versatility, fast reasoning, and easy expansion. Preliminary experiments show that the system can quickly infer and locate possible fault locations based on phenomena, and achieve basic fault diagnosis capabilities. With the continuous maturity of domain ontology application technology, ontology and rule-based reasoning diagnosis technology will be widely used in various equipment maintenance work, and gradually integrated into the intelligent transformation of various industries, and accelerate the pace of promoting the development of intelligent information in our country.

References

1. Zhou, A., Yu, D., & Zhang, W. (2015). A research on intelligent fault diagnosis of wind turbines based on ontology and fmeca.
2. Advanced Engineering Informatics, 29(1), 115-125.
3. Qiang, Z., Ping, Y., & Yang, X. (2017). Research on a knowledge modelling methodology for fault diagnosis of machine tools based on formal semantics. Advanced Engineering Informatics, 32(apr.), 92-112.
4. Pomeranz, I. R. S. M. (2010). On undetectable faults and fault diagnosis. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 29(11), 1832-1837.
5. Zhou, Q., Yan, P., Liu, H., Xin, Y., & Chen, Y. (2018). Research on a configurable method for fault diagnosis knowledge of machine tools and its application. The International Journal of Advanced Manufacturing Technology, 95(1-4), 937-960.
6. Leva, S., Mussetta, M., & Ogliari, E. (2019). Pv module fault diagnosis based on microconverters and day-ahead forecast. IEEE Transactions on Industrial Electronics, 66(5), 3928-3937.
7. Song, L. L., Wang, T. Y., Xu, L., & Song, X. W. (2016). Fault tree analysis of pantograph type current collector based on ontology modeling. Key Engineering Materials, 693, 1371-1376.

Published by: The Mattingley Publishing Co., Inc.