sensors

MDPI

Article

Ontology-Based Method for Fault Diagnosis of Loaders

Feixiang Xu, Xinhui Liu, Wei Chen *, Chen Zhou and Bingwei Cao

School of Mechanical Science and Engineering, Jilin University, Changchun 130022, China; xufeixiangsdut@163.com (F.X.); liuxh@jlu.edu.cn (X.L.); zhouchen17@mails.jlu.edu.cn (C.Z.); caobw16@mails.jlu.edu.cn (B.C.)

* Correspondence: chenwei_1979@jlu.edu.cn; Tel.: +86-186-4307-7790

Received: 21 December 2017; Accepted: 26 February 2018; Published: 28 February 2018

Abstract: This paper proposes an ontology-based fault diagnosis method which overcomes the difficulty of understanding complex fault diagnosis knowledge of loaders and offers a universal approach for fault diagnosis of all loaders. This method contains the following components: (1) An ontology-based fault diagnosis model is proposed to achieve the integrating, sharing and reusing of fault diagnosis knowledge for loaders; (2) combined with ontology, CBR (case-based reasoning) is introduced to realize effective and accurate fault diagnoses following four steps (feature selection, case-retrieval, case-matching and case-updating); and (3) in order to cover the shortages of the CBR method due to the lack of concerned cases, ontology based RBR (rule-based reasoning) is put forward through building SWRL (Semantic Web Rule Language) rules. An application program is also developed to implement the above methods to assist in finding the fault causes, fault locations and maintenance measures of loaders. In addition, the program is validated through analyzing a case study.

Keywords: loaders; fault diagnosis; ontology; CBR; RBR

1. Introduction

With the development of industry and technology, the modern engineering mechanisms, e.g., loaders, which are becoming increasingly complex in structural systems, hydraulic systems and electrical systems, have higher failure rates. Moreover, because of bad working circumstances, such as forest areas with slippery terrain, hilly areas and mine areas with rough terrain, etc., loaders have to confront different kinds of faults frequently, which make it difficult to find faults and diagnose faults in time [1]. In this context, it is urgent to study the loaders fault diagnosis method. Over the past decade, many methods have been developed in the fault diagnosis domain, for example, methods based on signal analysis, information knowledge-based methods, and model-based methods [2]. In particular, the model based on fault diagnosis is nowadays accepted as a powerful tool to solve fault detection problems in technical processes [3].

The model-based fault diagnosis method was originally developed by Beard, in 1971, to replace hardware redundancy by analytical redundancy [4], which has been developed for linear or linearized models. For example, Steven X. Ding et al. [5] proposed a data-driven framework for the design of observer-based fault detection and control systems. The linear model represents the system behavior only near one operating point. Indeed, when faults occur, the operating point changes; therefore, a linear model is not representative [6]. Thus, to consider an operating range of nonlinear systems, the Takagi-Sugeno (T-S) fuzzy model has been applied in the diagnosis of complex nonlinear systems [7]. The fault estimation observer was developed in [3] for discrete-time T-S fuzzy systems based on piecewise Lyapunov functions. The unknown inputs of observers are designed in a finite frequency domain for T-S fuzzy fault detection [8]. T. Youssef et al. [6] used a Proportional Integral observer design for actuator and sensor faults estimations based on the T-S fuzzy model. Although the fault diagnosis methods mentioned above can diagnose loaders' faults by using a small amount of real-time data, they require an explicit model representing the input–output relationship; the fault diagnosis performance of loaders relies on this model's accuracy [9]. Moreover, using the above methods, it is hard to present a universal approach for the fault diagnosis of all loaders. Different kinds of loaders have different components and design principles, resulting in various diagnostic methods. Even for the same type of loader, components might be quite different due to various factors, such as repair, customization or update. To solve these problems, case-based reasoning (CBR) is exploited, as does not need an explicit and accurate model. CBR is an important method used in learning and solving problems in artificial intelligence [10], which obtains the solution of the target case by visiting a source case that is similar to the target case in its case library based on past practical experiences [11,12]. In this context, the faults of all loaders can be diagnosed based on past experiences. What is more, its problem-solving ability increases with the accumulation of experience. One disadvantage of CBR is that the fault diagnosis will be invalid when there is no corresponding case. Therefore, the rule-based reasoning (RBR) method is proposed for diagnosing loaders' faults once CBR fails.

CBR and RBR fault diagnosis methods have the following shortcomings in the knowledge management of fault diagnosis.

(1) It is difficult for computers to understand and integrate fault diagnosis knowledge of loaders, especially when there are heterogeneous knowledge types, wide knowledge sources and different storage types of knowledge.
(2) It is difficult to reuse the existing loaders fault diagnosis model. Numerous scholars have to reestablish models and methods during the study of loaders fault diagnosis, resulting in the waste of research energy.

To overcome the two defects mentioned above, the concept of ontology was proposed, which is a mechanism that describes concepts and their systems' relationships [13,14]. As the basis of semantic knowledge, ontology can make concepts, vocabularies, attributes and other knowledge of the domain semantically, which enables the computer to easily understand and integrate the knowledge, thus sharing and reusing knowledge [15,16]. Hence, in this paper, we will exploit the ontology concept in our method.

In this paper, we combine ontology, CBR and RBR to realize the fault diagnosis of loaders, which assists users in finding the fault causes, fault locations and maintenance measures of loaders.

The rest of this paper is organized as follows: Section 2 briefly introduces the background and related works; Section 3 explains the overall structure of the fault diagnosis system; Section 4 focuses on the construction of the fault diagnosis ontology model; Section 5 describes the ontology-based CBR fault diagnosis method; Section 6 describes the ontology-based RBR fault diagnosis method; Section 7 describes the system implementation and validation; The final section contains the study's conclusions and suggestions for further work.

2. Background and Related Works

Previous research regarding loaders fault diagnosis methods mainly covers ontology and CBR/RBR methods. The following is the background and literature review of related works.

2.1. Ontology

Ontology has been applied in fault diagnosis, decision analysis, prediction and early warning in recent years. The ontology model is described as O = <C, OP, DP, I> [17,18], where C is the set of classes; OP is the set of object properties; DP is the set of data properties; and I is the set of instances.

OWL (Web Ontology Language) is an ontology description language, recommended by W3C (World Wide Web Consortium) international standards, which has a strong semantic ability to describe classes and attributes [19]. OWL1 is used to describe the ontology model in this paper, which consists of OWL Lite, OWL DL and OWL Full sublanguages. OWL DL is selected as the description language because it can incorporate description and reasoning. Sparql (Simple Protocol and RDF Query Language) is a language that enables query information in the OWL model [20]. The Jena inference engine can parse and reason ontology model. The combination of OWL, Jena inference engine and Sparql is the hot direction of knowledge representation [21]. Yu-Jun Wang et al. [22] described food cold chain quality ontology by using OWL and the Jena inference engine. The threshold of logistics parameters was queried, to obtain parameters' statuses based on Sparql. Xiao-Ci Huang et al. [23,24] described a water quality monitoring ontology model based on OWL and used the Jena inference machine to analyze the ontology model. Then, Sparql was used to query the threshold of water quality related parameters, so as to get the statuses of parameters.

With the rapid development of knowledge technology, ontology has been applied in areas like fault diagnosis. Rong Chen et al. [25] put forward an ontology-based knowledge modeling approach for fault diagnosis for rotating machinery. The ontology model was built to integrate fault diagnosis knowledge of rotating machinery. Furthermore, rules were added to allow the reasoning of fault diagnosis knowledge. However, this research did not deduce root causes for faults but described the reasons for general fault causes. In addition, the defined rules were relatively simple. Haizai Peng et al. [26] proposed a fault diagnosis method for conveyor-based ontology which was used to construct a knowledge basis for conveyor fault diagnosis. This study established the mapping relationships between fault phenomenon ontology, fault cause ontology and fault solution ontology, which have given us the guidance to manage knowledge representation well and independently through constructing sub-ontologies. G. Medina-Oliva et al. [27] proposed a fleet-wide approach based on ontologies in order to capitalize on knowledge and data to help decision makers to identify the causes of abnormal operations. The ontology modelling process in G. Medina-Oliva's research was similar to this paper, which used Protégé to construct ontology coded in OWL, and built Semantic Web Rule Language (SWRL) rules to diagnose faults. Alfonso Castro et al. [28] designed a multimedia service and resource management architecture for fault diagnosis. This architecture includes three types of automatic reasoning: heterogeneous, ontology-based and Bayesian reasoning. The combination of Bayesian reasoning and ontology-based reasoning performed better on incomplete datasets and in the presence of uncertainty, which could avoid the data loss problem caused by data acquisition devices in this paper. FD Samirmi et al. [29] aimed to develop an improved ontology model for transformer fault diagnosis by applying the fuzzy ontology. Similar to this paper, Protégé software and OWL DL were applied to build the ontology model. Based on ontology, fuzzy theory was introduced to solve uncertainty or imprecision problems for fault diagnosis.

2.2. CBR/RBR Method

In recent years, CBR and RBR have received more and more attention and have been applied for fault diagnosis as two main techniques in artificial intelligence and expert systems. Tomas, Gang Ma et al. [30] put forward an intelligent fault diagnosis method for power equipment by retrieving historical cases based on CBR. Olsson, T. et al. [31] classified the fault diagnoses and deduced the fault probabilities regarding CBR. Dong-yang Dou et al. [32] proposed an RBR method to achieve fault diagnosis for rotating machines. Xiao-Wen Deng et al. [33] designed an RBR based fault diagnosis expert system for wind turbines. Olsson, E. et al. [34] combined the acoustic signal and CBR to obtain the fault diagnosis of industrial robots. Dinh PhuocVo et al. [35] presented a hybrid knowledge-based system which proposed to a diagnosis approach for test engineers based on the combination of CBR and RBR to be used when an incident occurs.

However, the ability to understand knowledge semantically based on CBR and RBR is very poor [36,37]. As we all know, fault diagnosis based on CBR and RBR is usually a high intelligence activity, and experts often begin their diagnosis by observing, querying and interpreting faults diagnosis experience knowledge. In this way, experts form vocabularies of semantic expressions, but semantic ambiguity often occurs in experience observations and queries when experts do not know the exact expressions or the related concepts—though they may have some professional knowledge [36]. Hence, ontology is exploited to improve the ability to understand knowledge semantically based on CBR and RBR. Nadjette Dendani-Hadiby et al. [38,39] established an ontology model to obtain the semantics of knowledge, and then integrated CBR with ontology to develop a fault diagnosis system for steam turbines. An-mei Zhou et al. [40] combined an ontology model and the RBR method to obtain intelligent fault diagnosis for wind turbines, and the feasibility of the method was analyzed with practical cases. Based on an ontology model, Xiao-ci Huang et al. [41] designed custom rules to determine intelligent fault diagnoses for off-line Electronic Control Unit (ECU). Tung et al. [42] developed a solution retrieval system based on ontology, CBR and RBR. The experimental results show that this method can improve the accuracy of retrieval cases and reduce retrieval time prominently. Shao-li Chen et al. [37] built the disassembly knowledge model based on ontology, and then used CBR and RBR to obtain automated decision-making for the disassembly of mechanical products. Combining ontology and RBR, Zhou Qiang et al. [43] presented an intelligent fault diagnosis method for machine tools, and validation occurred by studying cases. D. Wang et al. [44] presented a new approach to the fault diagnosis of power transformers based on ontology and RBR. Based on the ontology and fault tree analysis methods, Wu Chun-yin [45] established a fault diagnosis knowledge model, and the CBR method was proposed to diagnose the faults of a tractor.

Inspired by previous work, as a novel attempt, ontology, CBR and RBR are applied into the field of loaders fault diagnosis in this paper.

3. The Overall Structure of Fault Diagnosis System

The main purpose of this paper is to design a loaders fault diagnosis system to help users find the fault causes, fault locations and fault maintenance measures of loaders in adequate time, based on ontology, CBR and RBR. Figure 1 shows the overall framework of the fault diagnosis system. The running parameters of the loaders are integrated in the data merging layer through data acquisition and transmission, based on sensor and embedded technologies, and then running parameters, the ontology model, the fault diagnosis case base and the fault diagnosis rule base are gathered in the data merging layer. Following this, fault diagnosis is achieved in combination with CBR and RBR; finally, fault diagnosis results are displayed on a user interface. As can be seen from Figure 1, the data collected by the sensor is the database of the whole fault diagnosis system. On one hand, in the CBR module, the collected data is used as one feature index of case-retrieval to match a suitable case in the case base. On the other hand, in the RBR module, the collected data is compared with data threshold values to judge instances of the *Parameters* class of ontology model, and then fault diagnosis is realized by combining ontology and defined rules. The function of each layer is described in detail.

- Data acquisition layer. Firstly, the sensors collect data (such as engine speed, oil pressure of engine, hydraulic oil temperature etc.) every 2 s. Secondly, the collected data is encapsulated by three ECU controllers, including engine ECU, gearbox ECU and hydraulic system ECU. Thirdly, the data acquisition device gathers and processes all data from the three ECU controllers according to the Controller Area Network (CAN) protocol every five minutes. On one hand, within 5 min, the data acquisition device processes 150 data from each parameter to obtain an average value for each parameter. In this way, the average values of running parameters for fault diagnosis are more useful and effective. On the other hand, data is transferred to the data merging layer every 5 min, with the aim of reducing the frequency of data transmission so as to decrease the data processing pressure on the data acquisition device and data merging layer. This data acquisition device is developed by embedded technology, and it is composed of an Microcontroller Unit

(MCU), a power module, an Liquid Crystal Display (LCD) module, a CAN module and a 3G module. MCU selects the stm32f103 microprocessor as the main control unit, which is responsible for receiving data, analyzing data and sending data. The power module mainly provides the power supply for the data acquisition device to ensure normal operation. The LCD module is used to display the operating data. The CAN module is used to receive data from three ECU controllers. The 3G module is responsible for sending data to the data merging layer. At last, the processed data is transferred to the data merging layer via the Internet in the form of Transmission Control Protocol/Internet Protocol (TCP/IP). The collected data lays the data foundation for the fault diagnosis layer to diagnose loaders' faults. The data package from each ECU controller has 29 bytes, and then the data merging layer transferred from the data acquisition device has 29 × 3 = 87 bytes. Given that the loader works 8 h per day, and the historical data can be saved for three months, the memory size for collected data is 87 × 8 × 60/5 × 30 = 250,560 bytes = 245 MB. Currently, there is one diagnosed loader; the memory size for collected data is 245 MB. Supposing that the ontology-based fault diagnosis system will diagnose 1000 loaders in the future, then the memory size for collected data is 245 MB × 1000/1024 = 239 GB.

- Data merging layer. The data merging layer is used to integrate collected data, the ontology model, the fault diagnosis case base, and the fault diagnosis rule base. The data collected by sensors is transferred to the data merging layer through the data acquisition device. The fault diagnosis ontology model semantically handles the relevant information of loaders fault diagnoses, so that the information can be stored and expressed. The case base summarizes previous diagnosis experience and provides case support for CBR. The rule base, which is defined by experts and expressed in the form of SWRL, provides rule support for RBR. The data merging layer is responsible for laying the foundation for the fault diagnosis layer, which is of great significance. Moreover, the ontology model, rule base and case base can be updated and improved continually. With the increase in data in the data merging layer, it will be able to diagnose more faults and can diagnose more accurately.
- Fault diagnosis layer. This layer is the core part for diagnosing loaders' faults. Based on the fault diagnosis ontology model, the CBR module is used for fault diagnosis following four steps: feature selection, case-retrieval, case-matching and case-updating. The CBR module is able to obtain accurate fault diagnoses because it exploits historical cases which have actually happened. Once the CBR module fails, due to the lack of corresponding cases, the RBR module will be executed. The RBR module is used to deduce fault causes, fault locations and maintenance measures based on defined SWRL rules which make up the fault diagnosis rule base. By making full use of the advantages of the ontology model in knowledge management, CBR and RBR can obtain accurate and effective fault diagnoses of loaders.
- UI (user interface). The UI was mainly developed for users, experts and administrators. Users (including drivers, operators and manufacturers) can obtain fault diagnosis results of loaders through UI. Experts are responsible for constructing and managing the case library and the rule library. Administrators mainly design and supervise the fault diagnosis ontology model. The results, including fault causes, fault locations and maintenance measures, can be obtained through UI. The UI application program was developed with the Visual Studio2017 software platform and C# language under the Windows 7 system.

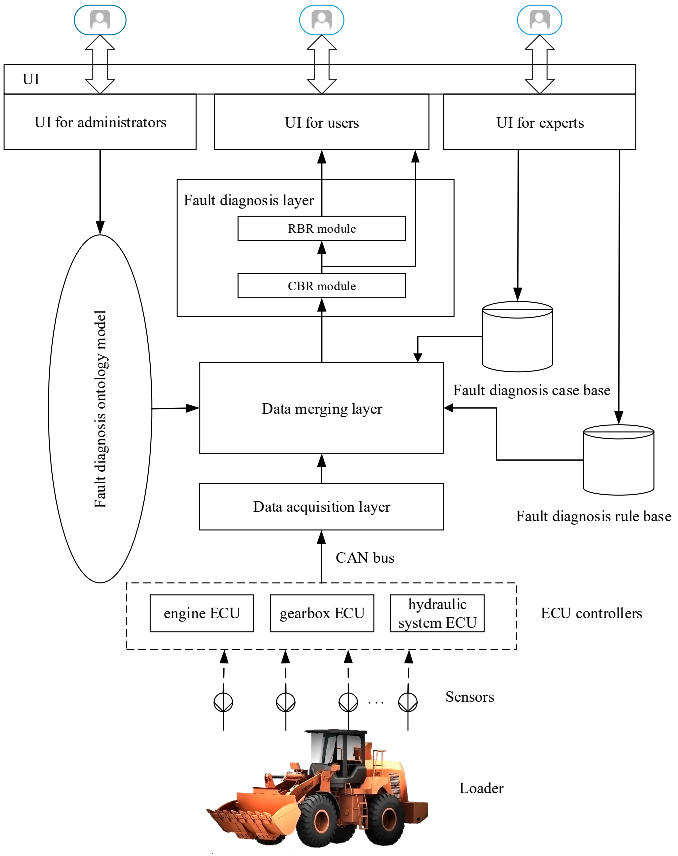


Figure 1. Overall structure of loaders fault diagnosis system. CBR, case-based reasoning; RBR, rule-based reasoning; UI, user interface; CAN, Controller Area Network; ECU, Electronic Control Unit.

4. Construction of the Fault Diagnosis Ontology Model

This section mainly introduces the fault diagnosis ontology model including ontology parsing and querying to lay the foundation for CBR/RBR.

4.1. Construction of the Ontology Model

Protégé [46] software was selected for building the fault diagnosis ontology model because of its user-friendly interface, powerful tools and data checking feature [47]. The ontology proposed in this paper is mainly used in the field of loaders fault diagnosis; thus, it can be classified as a highly specialized domain ontology. Currently, there are many kinds of approaches for constructing domain ontologies, i.e., METHONTOLOGY, SENSUS, TOVE, IDEF5 and the Seven-Step Method. Toshihiro Uchibayashi et al. [48] proposed a domain specific sub-ontology derivation end-user tool for the semantic grid, where ontology is built through extracting the sub-ontologies. This method is suitable for large-scale ontology domains and will be applied in this paper, when fault diagnosis ontology increases gradually. Hu qingxi et al. [49] built an ontology for representing the production process knowledge of workshops based on the IDEF5 method which has disadvantages in technical support. Zhou Yong [50] constructed an improved ontology model based on machine learning (BP neural network) which changed the mode of multi strategy merging in ontology mapping. S. Chaware et al. [51] integrated METHONTOLOGY and SENSUS to construct an ontology model for the shopping mall domain, where ontology can be prepared to get more correct information at a faster rate. However, this integrated method does not consider the collaborative and distributed construction of ontologies. The Seven-Step Method is among the top choices for building a domain ontology [22]. In addition, it not only enjoys detailed technical support and advantages in model creation, but also has advantages in the detailed modelling process [22]. Hence, this paper chose the Seven-Step Method, developed by the Medical Information Center of Stanford University [52], to construct the ontology model in the fault diagnosis domain, and it is shown in Figure 2. Next, the ontology modeling steps of the Seven-Step Method are described in detail.

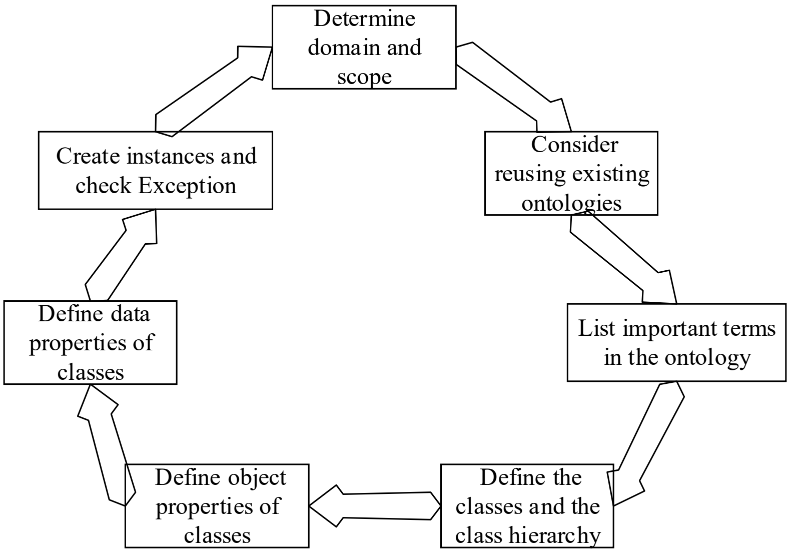

Determine domain and scope
Consider reusing existing ontologies
List important terms in the ontology
Define the classes and the class hierarchy
Define object properties of classes
Define data properties of classes
Create instances and check Exception

Figure 2. Seven-Step Method for ontology modeling.

- Determine the domain and scope

The research described in this paper is in the field of loaders fault diagnosis.

- Consider the reuse of existing ontologies

There is no suitable ontology that can be reused directly.

- List important terms in the ontology

The keywords of loaders fault diagnosis are engine, gearbox, oil temperature, gear pump, main valve, etc.

- Define classes and class hierarchy

The components of the loaders can be divided into four levels, from top to bottom, which are the device level, system level, assembly level, and part level. When the loader fails, a single fault can be aroused by various fault causes rather than one fault cause. Faults in low level components affect not only faults at same level, but also faults in higher levels. Thus, the causes of loaders' faults are very complex.

In order to handle the complexity of faults, the fault diagnosis ontology model has five defined classes: *FaultMode*, *FaultEquipment*, *FaultMaintenance*, *Parameters* and *FaultPhenomenon*, as shown in Figure 3. The *FaultMode* is composed of two subclasses: *FaultCause* and *FaultEffect* [40]. The *FaultEquipment* indicates the location of faults [43]. The *parameters* class expresses running data collected by sensors. The *FaultPhenomenon* indicates the phenomena when a failure occurs. The *FaultMaintenance* means that faults can be repaired by some measure.

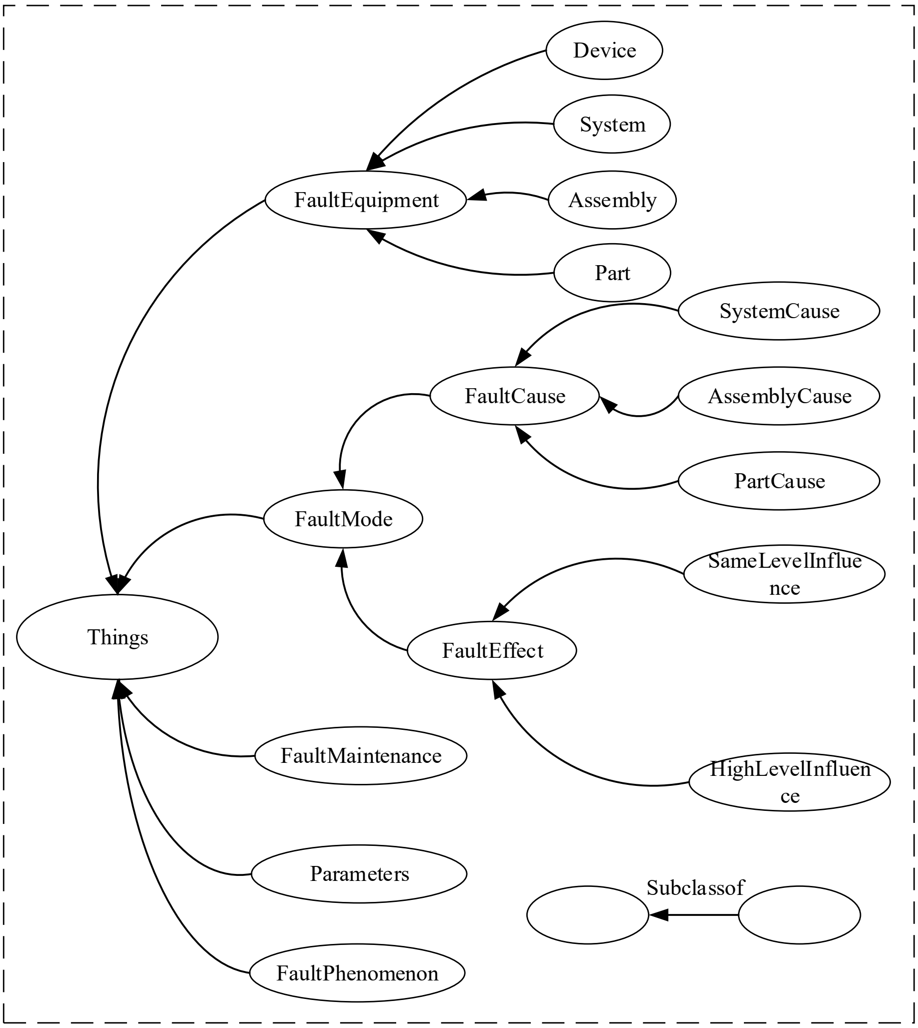


Figure 3. Classes in the fault diagnosis ontology model.

- Define object properties of classes

As shown in Table 1, each object property has its corresponding domains and ranges. If an object property has an inverse property, then the inverse property has inverse domains and ranges.

(1) Property 1 indicates the change in operating parameters caused by the fault mode.
(2) Property 2 shows the relationship between the fault mode and the fault causes.
(3) Property 3 indicates the relationship between the fault components and their sub-components.
(4) Property 4 shows the relationship between the fault mode and fault effects. Property 5 and 6 are sub-properties of property 4, and property 5 indicates the relationship fault mode and its same level effects, and property 6 indicates the relationship fault mode and its higher level effects.
(5) Property 7 indicates that fault locations provide information support for maintenance methods.
(6) Property 8 indicates that the fault mode is able to be repaired through using maintenance methods.
(7) Property 9 shows that the fault mode happens in components of loaders.
(8) Property 10 indicates that the fault mode is accompanied by phenomena.

Table 1. Object properties in the fault diagnosis ontology.

No	Object Property	Domains	Ranges	Inverse Property
1	*stemFrom*	*FaultMode*	*Parameters*	*leadTo*
2	*causeIs*	*FaultMode*	*FaultMode*	N/A
3	*belongTo*	*FaultEquipment*	*FaultEquipment*	N/A
4	*toEffect*	*FaultMode*	*FaultMode*	N/A
5	*toSameLevelEffect*	*FaultMode*	*FaultMode*	N/A
6	*toHighLevelEffect*	*FaultMode*	*FaultMode*	N/A
7	*offerInformation*	*FaultEquipment*	*FaultMaintenance*	N/A
8	*toMaintenance*	*FaultMaintenance*	*FaultMode*	*maintenanceBy*
9	*happenAt*	*FaultMode*	*FaultEquipment*	*hasFault*
10	*hasPhenomenon*	*FaultMode*	*FaultPhenomenon*	*originateFrom*

- Define data properties of classes

As shown in Table 2, each data property, similar to the object properties, also has its own domains and ranges.

Table 2. Data properties in the fault diagnosis ontology.

No	Data Property	Domains	Ranges	Description
1	*hasNumber*	*FaultEquipment*	string	To record types of components
2	*madeFactory*	*FaultEquipment*	string	To indicate the manufacturer that produces the components
3	*madeTime*	*FaultEquipment*	datetime	To indicate the production time of components
4	*hasMaxValue*	*Parameters*	float	To indicate the maximum value in the normal range of parameters
5	*hasMinValue*	*Parameters*	float	To indicate the minimum value in the normal range of parameters
6	*monitoringTime*	*Parameters*	datetime	To record the monitoring time of parameters
7	*memoryValue*	*FaultMode*	float	To record the memory value of a fault in case library, also used for case-updating

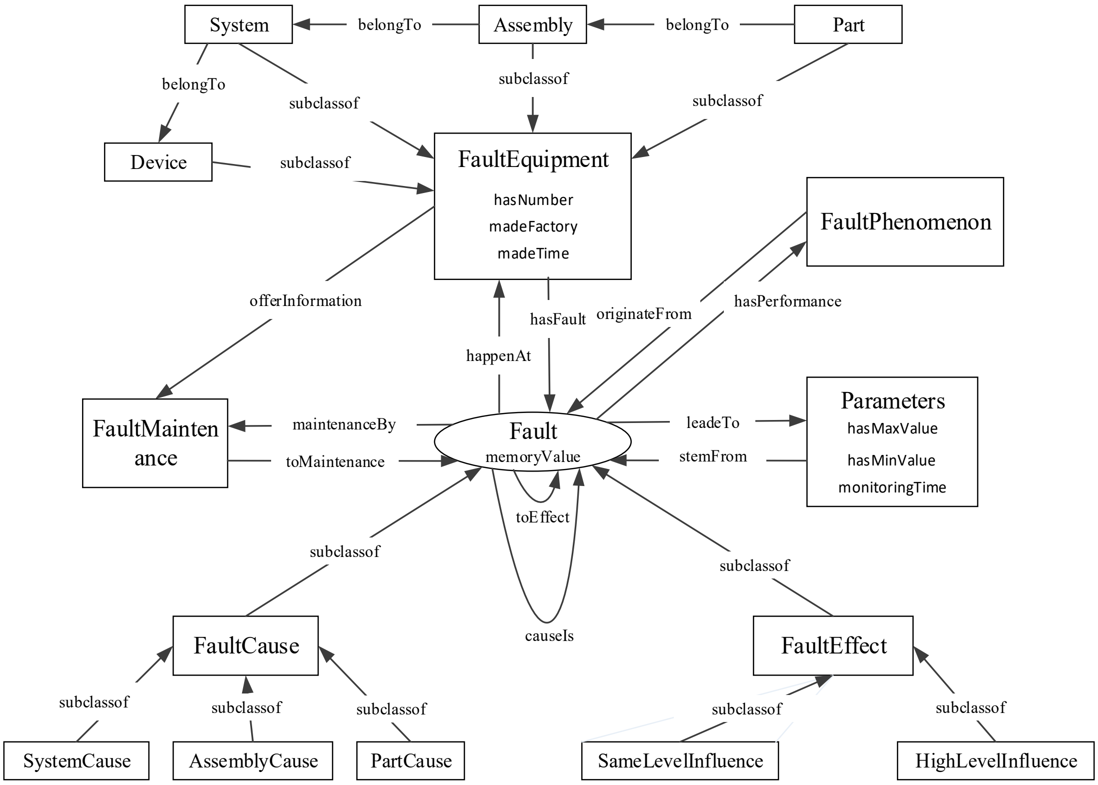


Figure 4. Structure of the fault diagnosis ontology model.

After the former six steps, the structure of the fault diagnosis ontology model is built, as shown in Figure 4.

- Create instances and check exceptions

Instances of fault diagnosis ontology model are divided into two parts. For the first part, instances of the parameters' classes and their related properties are filled up by data from the loader's data acquisition device. For the second part, system administrators and experts are responsible for instantiating other classes and properties according to the actual condition of the loaders. So far, the loaders fault diagnosis ontology model has been established after the instantiation. A pellet is used for checking exceptions, to verify the correctness of ontology model [22].

4.2. Parsing and Querying of the Ontology Model

After the fault diagnosis ontology model was constructed by Protégé software and described by OWL language, it needed to be parsed and queried to apply ontology model into the CBR/RBR method. As introduced in the second section, the Jena inference engine was used to parsethe ontology model in this paper, and the SELECT mode of the Sparql was selected to query the properties of the ontology model.

5. CBR for Loaders Fault Diagnosis

CBR, proposed by Schank (1983) [53], can simulates human cognitive processes and integrates empirical knowledge of different fields into a unified format [54]. CBR can achieve effective and accurate fault diagnosis of loaders. This is because that CBR can obtain causes of failure, fault locations and maintenance methods, as long as cases are able to be matched successfully in the case library. On the other hand, CBR can exploit historical cases which have actually happened and have been solved, to guarantee accuracy.

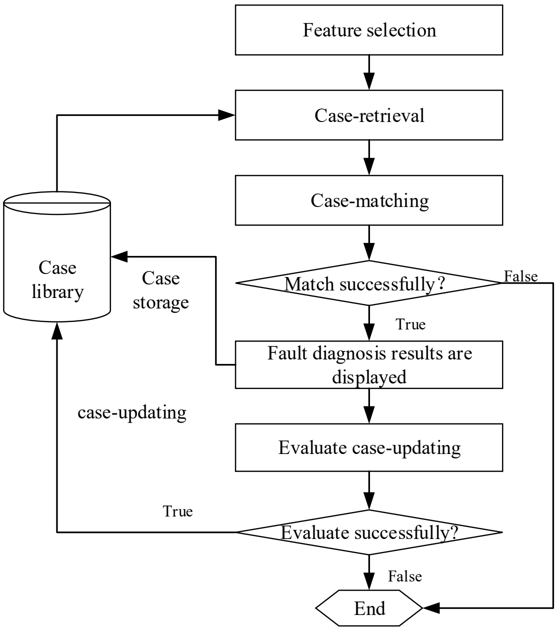


Figure 5. Process of the CBR based loaders fault diagnosis.

The process of CBR based loaders fault diagnosis is shown in Figure 5. Loader type, fault phenomena and parameters are selected as the feature indexes for calculating case similarities in the case library. When the similarity value is greater than or equal to the set value, it indicates that the case-matching has been successful, and the fault diagnosis results of the matched case are displayed directly; Then case-updating is evaluated, and the case library will be updated if the evaluation is satisfactory, or else the CBR method will end. When the similarity value is less than the set value, the CBR based fault diagnosis will fail and the diagnosis process will end. The fault diagnostic process is described in detail below.

5.1. Feature Selection and Case-Retrieval

Although there are a large number of loader types, repair methods are often similar for similar faults of the same loader type; therefore, the loader type is selected as an essential feature index. The maintenance methods used in the same fault phenomena can also be used as guidance to the maintenance person; thus, the fault phenomena index is another alternative. In addition, parameters obtained directly from the data acquisition device will change when faults occur, therefore, meeting the needs of the feature index. Hence, the three feature indexes are loader type, fault phenomena and operating parameters.

Case-retrieval refers to searching the fault diagnosis case library based on the above three feature indexes. Hence, the construction of the case library is of great significance. As shown in Figure 6, experts select the classes and properties of fault diagnosis ontology manually, and the corresponding data is filed manually regarding historical cases and events, and then fault diagnosis cases are built and stored in the case library. At this point, the case library is constructed. Moreover, the established ontology model with its semantic description can be retrieved with cases that are also built semantically in the above process of constructing the case base.

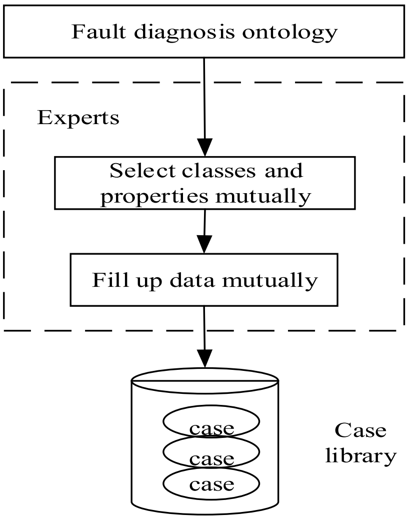


Figure 6. Process of constructing case library.

5.2. Case-Matching

The established ontology model with its semantic description can be retrieved with cases which are also built semantically, as mentioned above. Hence, case matching can be realized by matching instances of *hasNumber*, *FaultPhenomenon*, *Parameters* corresponding to three feature indexes between ontology models and cases. Similarity values are used to judge the similarity between to-be-diagnosed faults and the cases in the case library. The nearest neighbor algorithm [55] was used to calculate the similarity value in this paper, and its formula is shown in Equation (1).

$$Sim(C_i) = \sum_{j=1}^{m} (w_j * Sim(C_{ij})) \tag{1}$$

where I is the sequence number of the case in the case library. $Sim(C_i)$ is the similarity value between to-be-diagnosed faults and the *i*th case. J is the sequence number of the feature index. m represents the numbers of all feature indexes. w_j is the weight of the *j*th feature index. $Sim(C_{ij})$ is the similarity value between the *j*th feature of the *i*th case and the to-be-diagnosed faults.

The threshold of $Sim(C_i)$ is set as $Sim(C)$, which means that the case-matching is invalid when $Sim(C_i) < Sim(C)$; otherwise case-matching is successful. The value, w_j, is directly defined by system experts to simplify the designing of this value. $Sim(C_{ij})$ is determined as follows in this paper:

(1) If the loader type is the same as the case in the case library, then $Sim(C_{i1})$ equals 1, or else $Sim(C_{i1})$ equals 0.

(2) If fault phenomena are identical to a case in the case base, then $Sim(C_{i2})$ equals 1, or else $Sim(C_{i2})$ equals 0.

(3) Operating parameters collected by sensors are calculated with Equation (2).

$$Sim(C_{i3}) = \sum_{j=1}^{N} (m_k * Sim(f_k)) \tag{2}$$

where m_k represents the weight of the *j*th operating parameter, which is determined directly by system experts, N stands for the number of operating parameter, $Sim(f_k)$ is calculated by Equation (3), which represents the *j*th operating parameter similarity value between to-be-diagnosed faults and the case, p_k denotes the *k*th operating parameter value of to-be-diagnosed faults, and l_k represents the *k*th operating parameter value of the case.

$$Sim(f_k) = 1 - \frac{p_k - l_k}{p_k} \tag{3}$$

5.3. Case-Updating

When case-matching is achieved successfully, the actual fault diagnosis results can be saved into the case library. However, too many similar cases will waste the storage resource and decrease the efficiency of case-retrieval and case-matching. Therefore, it is necessary to develop case-updating strategy to eliminate redundancy in the case library. Derbinsky et al. [56] proposed a case update method which forgets outdated cases and reinforces the memory of frequent cases. Based on this principle, the memory value (M) of each case has the following relationship:

$$M = \frac{\pi}{2} - \arctan\Delta T, \Delta T \geq 0 \tag{4}$$

where M stands for the value of the *memoryValue* data property in the fault diagnosis ontology, ΔT indicates the existence time of the case in the case base, and its initial value is set as "0".

When cases are successfully matched, one of these cases is randomly retained, and its ΔT is set as "0"; then, the other cases are discarded. If we set the threshold of M to Mt, then the case is outdated and will be abandoned when $M < Mt$. By controlling suitable values of Mt, we can ensure that the case base does not become too narrowed down, oversized or skewed. In this way, a suitable number of cases can be saved in the case library. Additionally, cases in the case base can also be improved by different experts at any time to improve the effectiveness of case management. ΔT of the modified case will be reinitialized to "0".

6. RBR for Loaders Fault Diagnosis

Based on the fault diagnosis ontology model, RBR was used to process the loaders fault diagnosis when the CBR method failed due to the lack of proper cases.

6.1. Fault Diagnosis Rules

The standard SWRL language is used to express the fault diagnosis rules. Based on the analysis of classes and properties in the fault diagnosis ontology model, as shown in Table 3, the constructed SWRL rules are represented in this paper, and these rules are the basis of the ontology-based RBR method. To understand SWRL easily, two basic elements of syntax are introduced, as follows [57]:

(1) C (?x): If x is an instance of the class C or the value of its data property, then C (?x) is established;
(2) P (?x, y): If x and y are associated with the property, P, then P (?x, y) is valid.

Table 3. Fault diagnosis rules.

No	Rules
Rule 1	*FaultMode*(?x) ˆ*FaultPhenomenon*(?y) ˆ *hasPhenomenon*(?x, ?y) → *FaultEquipment*(?z) ˆ *happenAt*(?x,?z)
Rule 2	*FaultMode*(?x) ˆ *Parameters*(?y) ˆ *stemFrom*(?x, ?y) ˆ *Device*(?a) ˆ *happenAt*(?x, ?a) ˆ *FaultMode*(?z) ˆ *toHighLevelEffect*(?z, ?x) → *causeIs*(?x, ?z) ˆ *SystemCause*(?x)
Rule 3	*FaultMode*(?x) ˆ *Parameters*(?y) ˆ *stemFrom*(?x, ?y) ˆ *System*(?a) ˆ *happenAt*(?x, ?a) ˆ *FaultMode*(?z) ˆ *toHighLevelEffect*(?z, ?x) → *causeIs*(?x,?z) ˆ *AsseamblyCause*(?x)
Rule 4	*FaultMode*(?x) ˆ *Parameters*(?y) ˆ *stemFrom*(?x, ?y) ˆ*Asseambly*(?a) ˆ *happenAt*(?x, ?a) ˆ *FaultMode*(?z) ˆ *toHighLevelEffect*(?z, ?x) → *causeIs*(?x, ?z) ˆ *PartCause*(?x)
Rule 5	*FaultMode*(?x) ˆ *Parameters*(?y) ˆ *stemFrom*(?x, ?y) ˆ *System*(?a) ˆ *happenAt*(?x, ?a) ˆ *FaultMode*(?z) ˆ *toSameLevelEffect*(?z, ?x) → *causeIs*(?x, ?z) ˆ *SystemCause*(?x)
Rule 6	*FaultMode*(?x) ˆ *Parameters*(?y) ˆ *stemFrom*(?x, ?y) ˆ *Asseambly* (?a) ˆ *happenAt*(?x, ?a) ˆ *FaultMode*(?z) ˆ *toSameLevelEffect*(?z, ?x) → *causeIs*(?x, ?z) ˆ *Asseambly Cause*(?x)
Rule 7	*FaultMode*(?x) ˆ *Parameters*(?y) ˆ *stemFrom*(?x, ?y) ˆ*Part*(?a) ˆ *happenAt*(?x, ?a) ˆ *FaultMode*(?z) ˆ *toSameLevelEffect*(?z, ?x) → *causeIs*(?x, ?z) ˆ *PartCause*(?x)
Rule 8	*FaultMode*(?x) ˆ *FaultEquipment*(?y) ˆ *happenAt*(?x,?y) ˆ *PartCause*(?x) → *Part*(?y)
Rule 9	*FaultMode*(?x) ˆ *FaultEquipment*(?y) *happenAt* (?x,?y) ˆ *AsseamblyCause*(?x) → *Asseambly*(?y)
Rule 10	*FaultMode*(?x) ˆ *FaultEquipment*(?y) ˆ *happenAt*(?x,?y) ˆ *SystemCause*(?x) → *System*(?y)
Rule 11	*FaultMode*(?x) ˆ *FaultEquipment*(?y) ˆ *happenAt*(?x,?y) ˆ *FaultMaintenance*(?z) ˆ *offInformation*(?y,?z) → *toMaintenance*(?z, ?x)

In order to simplify the discussion here, we will only explain the following representative rules.

- Rule 1: If fault mode x occurs with the occurrence of fault phenomena y, then the direct fault locations of fault mode x are z.
- Rule 2: If fault mode x's change derives from parameter y, and x happens at device component a, and fault mode z affects x at the same level, then the cause of x is fault mode z, and x has fault causes at the system level.
- Rule 5: If fault mode x's change derives from parameter y, and x happens at system component a, and fault mode z affects x at a higher level, then the cause of x is fault mode z, and x has fault causes at the system level.
- Rule 8: If fault mode x happens at equipment y, and the reasons for the components' faults are x, then the fault locations are y.
- Rule 11: If the fault mode x happens at equipment y, and the fault locations provide the method for troubleshooting z, then failure maintenance z can repair fault mode x.

6.2. RBR for the Loaders Fault Diagnosis Process

The process of loaders fault diagnosis based on RBR is shown in Figure 7. Firstly, the direct fault locations are deduced based on Rule 1 and the *FaultPhenomenon* property of fault diagnosis ontology.

Secondly, Rules 2~7 are triggered to determine all probable fault causes by combining *stemFrom*, *happenAt* and *toEffect* from high to low levels. Rule 2 and Rule 5 are responsible for reasoning fault causes at the system level. Rule 3 and Rule 6 are triggered to determine fault causes at the assembly level. Rule 4 and Rule 7 are triggered to reason fault causes at the part level.

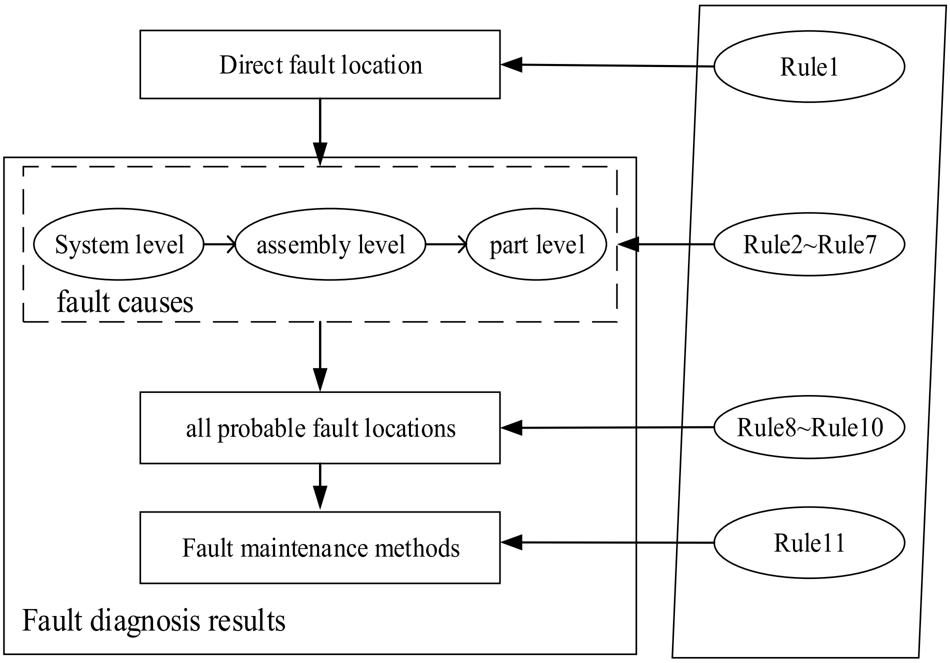


Figure 7. Process of loaders fault diagnosis based on RBR.

Thirdly, Rules 8~10 are triggered to deduce all possible fault locations based on the *happenAt* object property. Finally, in accordance with Rule 11, maintenance methods can be obtained by integrating the *offerInformation* and *happenAt* object properties.

Overall, combining fault diagnosis ontology and custom rules defined in Table 3, fault diagnosis results, including fault causes, all probable fault locations and fault maintenance methods, can be deduced to users.

7. System Implementation and Validation

7.1. System Implementation

This section, describes the implementation of the loaders fault diagnosis system, in accordance with the aforementioned proposed methods. As shown in Figure 8, the system is divided into three parts: the development of the data acquisition device, the construction of the ontology model and the development of the application program. Sensors collect the running parameters, and then data is collected and packed by the data acquisition device. At last, data is transmitted to the remote Visual Studio2017 software development platform through the 3G communication module. The data collected by sensors is of great significance. In the CBR process, the collected data is used to calculate Equation (2). In the RBR module, the collected data is compared with data threshold values to judge instances of the *Parameters* class.

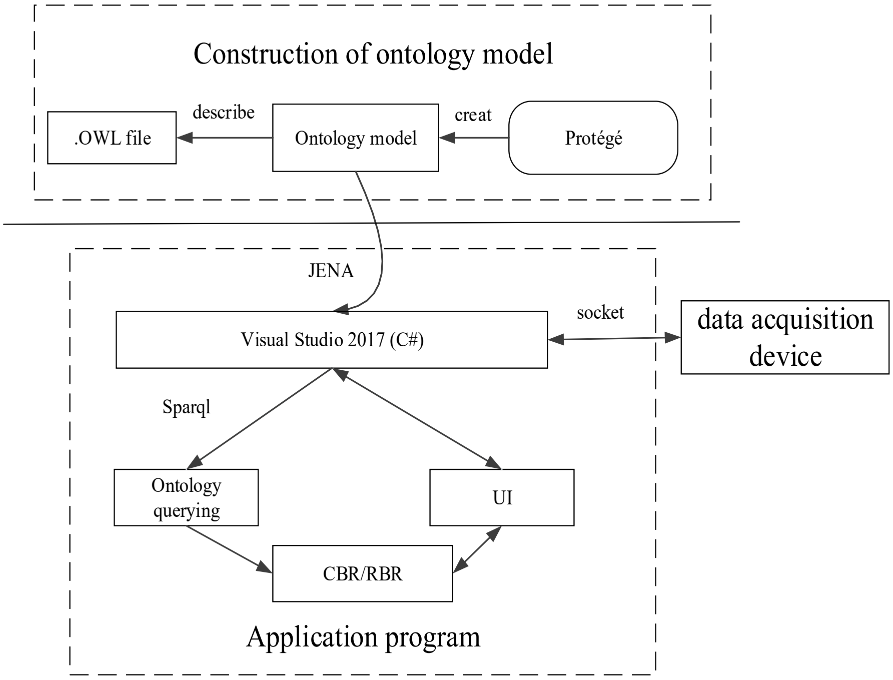


Figure 8. System implementation method for the fault diagnosis of loaders.

The fault diagnosis ontology model is instantiated by the actual loaders and collected parameters, and then it is built by Protégé software and described by OWL language.

The application program is developed by the Visual Studio2017 software platform and C# language under the Windows 7 system. The socket communication and multithreading technology are used to receive data from the data acquisition device. Jena packages and Java packages are imported into the application program, and then the Jena engineering is used for parsingthe fault diagnosis ontology. The SELECT mode of Sparql is responsible for querying the ontology model to read all properties. CBR and RBR are carried out according to the fifth and sixth chapters in this paper. So far, the loaders fault diagnosis system has been developed.

7.2. System Validation

This section uses the FW50GL wheel loader as an example to verify the validation of the system. As shown in Figure 9, the FW50GL wheel loader is divided into actual components to instantiate the device, system, assembly and part four subclasses in the ontology model, based on layered method mentioned in Section 4.1. Since there are so many components, we mainly describe the major components of the loaders.

Firstly, as shown in Figure 10, the data acquisition device installed in the cab of the FW50GL wheel loader mainly encapsulates the operating data collected by the sensors. As shown in Table 4, the operating parameters' names and threshold values of their states are translated into the ontology instances and their data properties, *hasMaxValue* and *hasMinValue*. Every parameter has three instances, for example, instances of engine oil temperature include *engine_oil_temperature_lower_state*, *engine_oil_temperature_noraml_state*, and *engine_oil_temperature_higher_state*. Three instances of each parameter are judged by comparing the data collected by sensors with data threshold values shown in Table 4. For example, when the engine oil temperature collected by a temperature sensor is 60 °C, then the instance of the class, *engine oil temperature*, is *engine_oil_temperature_normal_state*, according to the data threshold values (normal state: 40–120 °C) shown in Table 4.

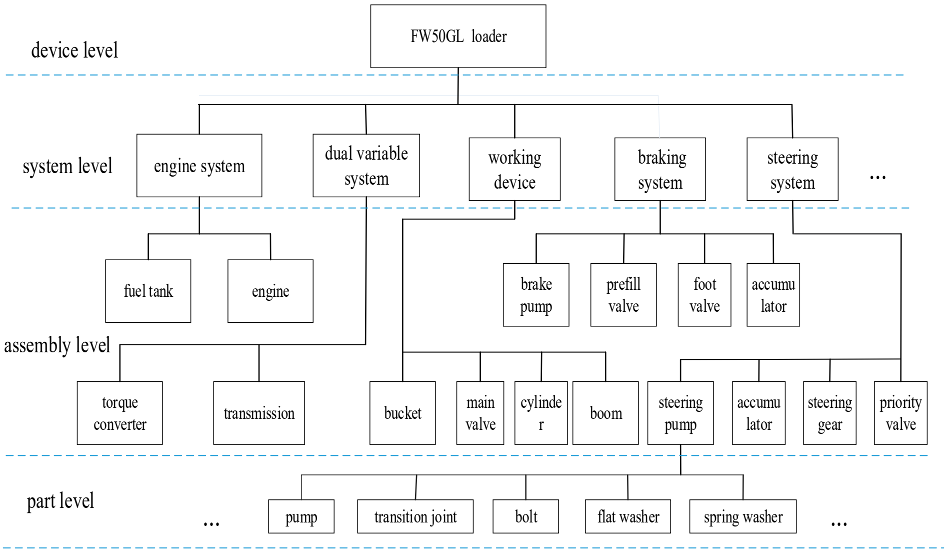


Figure 9. Components of the FW50GL wheel loader.

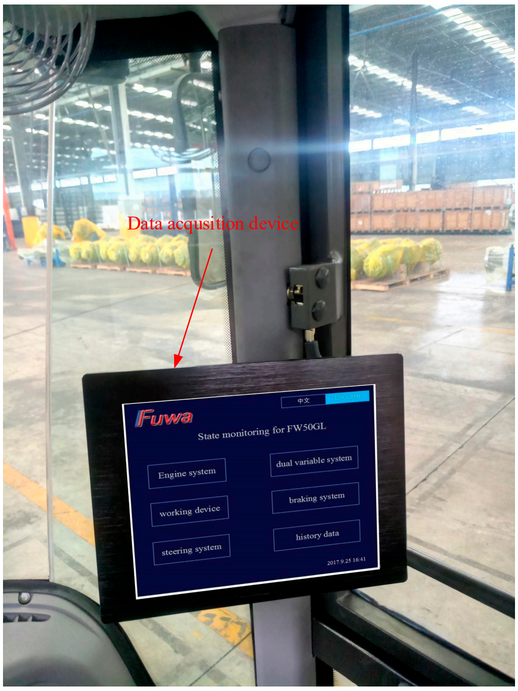


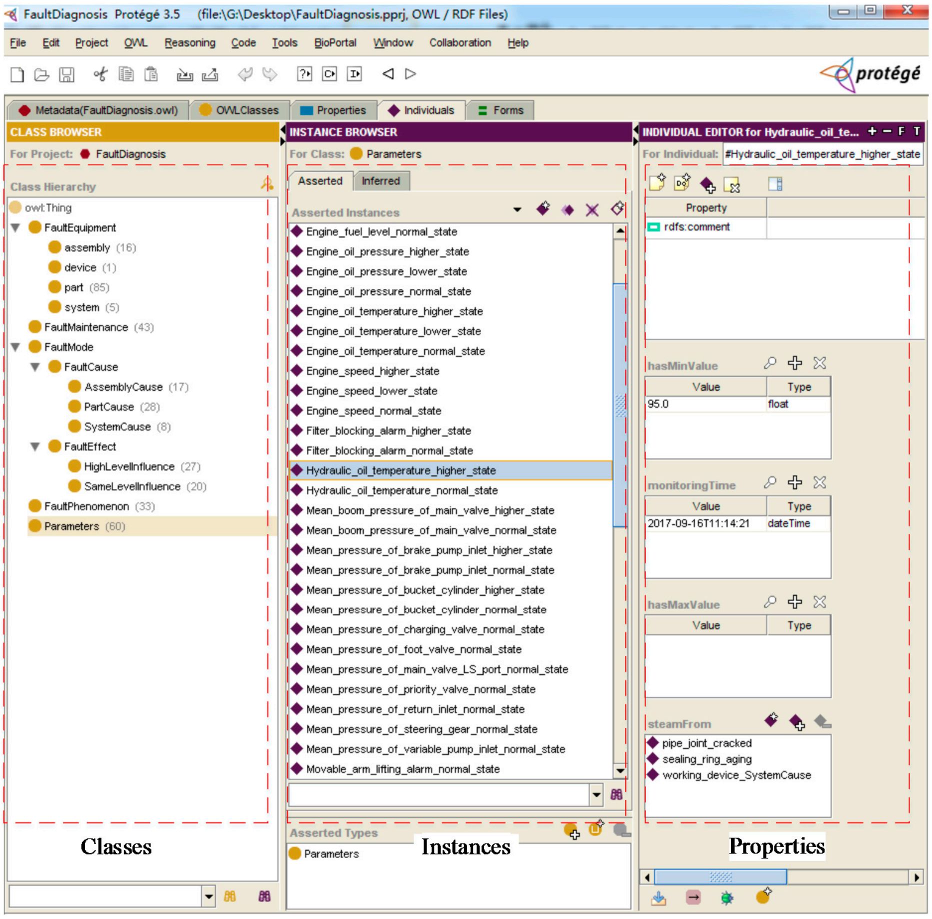
Figure 10. Data acquisition device installed in the cab of the FW50GL wheel loader.

Secondly, Protégé software is used to build the fault diagnosis ontology model, as shown in Figure 11. Classes, instances and properties are listed in order from left to right in Figure 11. Thirdly, the data merging layer is constructed through integrating collected operating parameters data, the built fault diagnosis ontology model, the fault diagnosis rule base and the fault diagnosis case base.

At last, an application program is developed in this paper to verify the CBR and RBR loaders fault diagnosis methods. The CBR loaders fault diagnosis results are shown in Figure 12. The similarity of case matching is calculated to be 0.8 according to the fault phenomena, loader type and parameters' feature indexes. This shows that the case matching is successful, and the fault diagnosis results of the matched case are displayed in the application program directly. The RBR loaders fault diagnosis results are shown in Figure 13. This shows that there is no suitable case in the case base when the similarity of case matching is 0.2. The RBR method is carried out to diagnose faults based on defined SWRL rules. Compared with actual failures and diagnostic results in the application program, it is demonstrated that the ontology-based loaders fault diagnosis method is reasonable and effective, and the developed fault diagnosis system can operate correctly.

Table 4. Operating parameters and threshold values of their states.

No	Parameters	Abnormal State 1 below Normal Value	Normal State	Abnormal State 2 above Normal Value
1	Engine speed	≤680 r/min	680–2400 r/min	≥2400 r/min
2	Engine oil pressure	≤0.07 Mpa	0.07–0.4 Mpa	≥0.4 Mpa
3	Engine coolant temperature	≤0 °C	0–110 °C	≥110 °C
4	Engine oil temperature	≤40 °C	40–120 °C	≥120 °C
5	Engine fuel level	≤6.25 L	6.25–50 L	≥50 L
6	Transmission oil temperature	≤0 °C	0–127 °C	≥127 °C
7	Transmission oil pressure	≤0.36 Mpa	0.36–2.24 Mpa	≥2.24 Mpa
8	Filter blocking alarm	>0	≤0 (0:normal 1:alarming)	>0
9	Movable arm lifting alarm	>0	≤0 (0:normal 1:alarming)	>0
10	Hydraulic oil temperature	≤0 °C	0–95 °C	≥95 °C
11	Mean pressure of variable pump inlet	≤0 Mpa	0–3 Mpa	≥3 Mpa
12	Mean pressure of steering gear	≤18 Mpa	18–21 Mpa	≥21 Mpa
13	Mean pressure of priority valve	≤22 Mpa	22–24 Mpa	≥24 Mpa
14	Mean pressure of bucket cylinder	≤22 Mpa	22–24 Mpa	≥24 Mpa
15	Mean boom pressure of main valve	≤22 Mpa	22–24 Mpa	≥24 Mpa
16	Mean pressure of main valve LS port	≤22 Mpa	22–24 Mpa	≥24 Mpa
17	Mean pressure of foot valve	≤6 Mpa	6–7 Mpa	≥7 Mpa
18	Mean pressure of brake pump inlet	≤0 Mpa	0–1 Mpa	≥1 Mpa
19	Mean pressure of charging valve	≤15 Mpa	15–18 Mpa	≥18 Mpa
20	Mean pressure of return inlet	≤0 Mpa	0–1 Mpa	≥1 Mpa



For Project: FaultDiagnosis

Class Hierarchy

- owl:Thing
- FaultEquipment
  - assembly (16)
  - device (1)
  - part (85)
  - system (5)
- FaultMaintenance (43)
- FaultMode
  - FaultCause
    - AssemblyCause (17)
    - PartCause (28)
    - SystemCause (8)
  - FaultEffect
    - HighLevelInfluence (27)
    - SameLevelInfluence (20)
- FaultPhenomenon (33)
- Parameters (60)

For Class: Parameters

Asserted Instances

- Engine_fuel_level_normal_state
- Engine_oil_pressure_higher_state
- Engine_oil_pressure_lower_state
- Engine_oil_pressure_normal_state
- Engine_oil_temperature_higher_state
- Engine_oil_temperature_lower_state
- Engine_oil_temperature_normal_state
- Engine_speed_higher_state
- Engine_speed_lower_state
- Engine_speed_normal_state
- Filter_blocking_alarm_higher_state
- Filter_blocking_alarm_normal_state
- Hydraulic_oil_temperature_higher_state
- Hydraulic_oil_temperature_normal_state
- Mean_boom_pressure_of_main_valve_higher_state
- Mean_boom_pressure_of_main_valve_normal_state
- Mean_pressure_of_brake_pump_inlet_higher_state
- Mean_pressure_of_brake_pump_inlet_normal_state
- Mean_pressure_of_bucket_cylinder_higher_state
- Mean_pressure_of_bucket_cylinder_normal_state
- Mean_pressure_of_charging_valve_normal_state
- Mean_pressure_of_foot_valve_normal_state
- Mean_pressure_of_main_valve_LS_port_normal_state
- Mean_pressure_of_priority_valve_normal_state
- Mean_pressure_of_return_inlet_normal_state
- Mean_pressure_of_steering_gear_normal_state
- Mean_pressure_of_variable_pump_inlet_normal_state
- Movable_arm_lifting_alarm_normal_state

Asserted Types

- Parameters

For Individual: #Hydraulic_oil_temperature_higher_state

Property	
rdfs:comment	

hasMinValue

Value	Type
95.0	float

monitoringTime

Value	Type
2017-09-16T11:14:21	dateTime

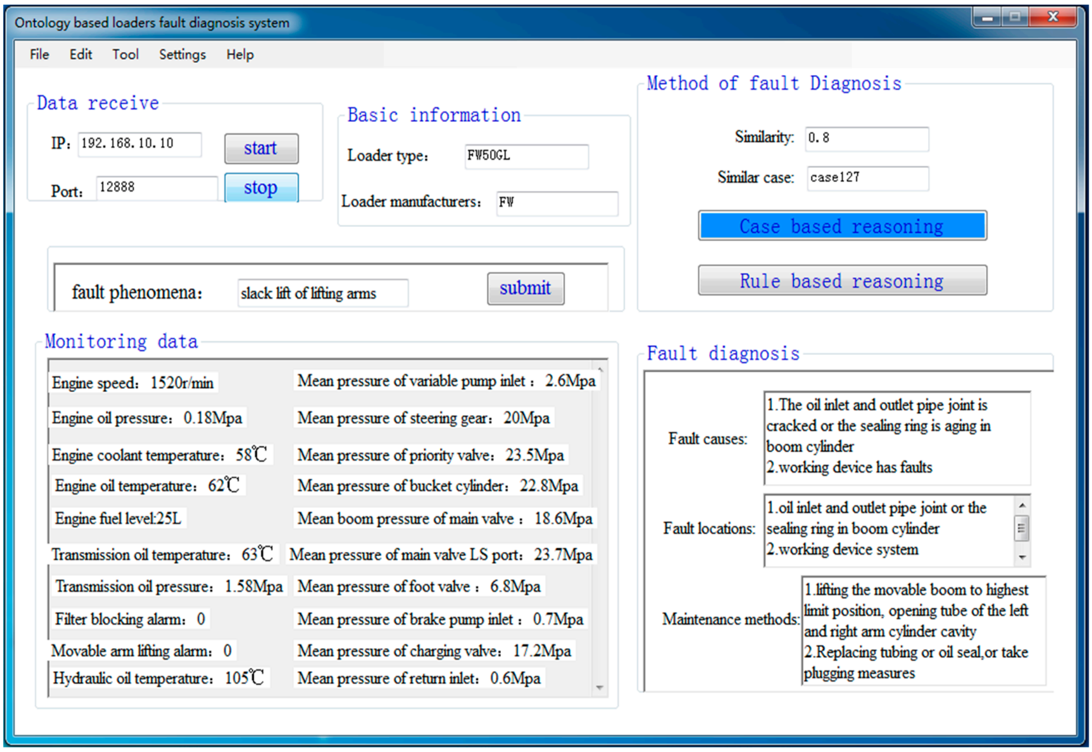
hasMaxValue

Value	Type

steamFrom

- pipe_joint_cracked
- sealing_ring_aging
- working_device_SystemCause

Figure 11. Fault diagnosis ontology model in the Protégé software.



IP：192. 168. 10. 10
Port：12888
Loader type：FW50GL
Loader manufacturers：FW
Similarity: 0. 8
Similar case: case127
fault phenomena: slack lift of lifting arms

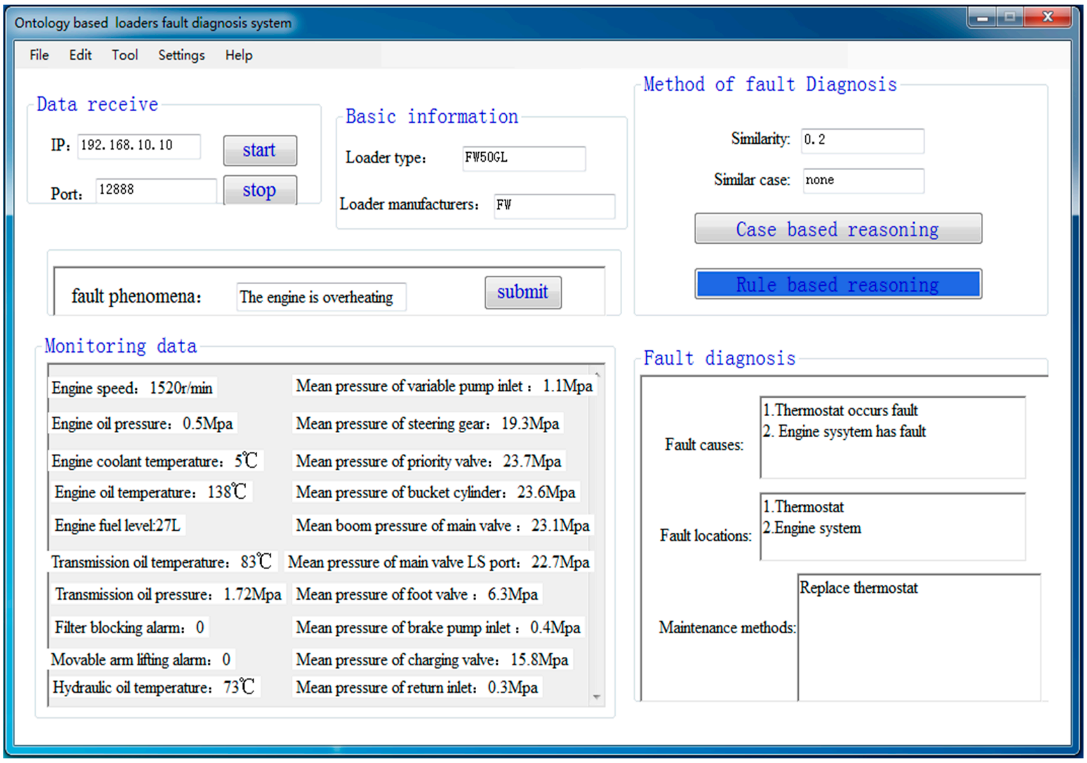
Engine speed：1520r/min
Mean pressure of variable pump inlet：2.6Mpa
Engine oil pressure：0.18Mpa
Mean pressure of steering gear：20Mpa
Engine coolant temperature：58℃
Mean pressure of priority valve：23.5Mpa
Engine oil temperature：62℃
Mean pressure of bucket cylinder：22.8Mpa
Engine fuel level:25L
Mean boom pressure of main valve：18.6Mpa
Transmission oil temperature：63℃
Mean pressure of main valve LS port：23.7Mpa
Transmission oil pressure：1.58Mpa
Mean pressure of foot valve：6.8Mpa
Filter blocking alarm：0
Mean pressure of brake pump inlet：0.7Mpa
Movable arm lifting alarm：0
Mean pressure of charging valve：17.2Mpa
Hydraulic oil temperature：105℃
Mean pressure of return inlet：0.6Mpa

Fault causes: 1.The oil inlet and outlet pipe joint is cracked or the sealing ring is aging in boom cylinder
2.working device has faults

Fault locations: 1.oil inlet and outlet pipe joint or the sealing ring in boom cylinder
2.working device system

Maintenance methods: 1.lifting the movable boom to highest limit position, opening tube of the left and right arm cylinder cavity
2.Replacing tubing or oil seal,or take plugging measures

Figure 12. CBR loaders fault diagnosis results.



IP：192. 168. 10. 10
Port：12888
Loader type：FW50GL
Loader manufacturers：FW
Similarity: 0. 2
Similar case: none
fault phenomena: The engine is overheating

Engine speed：1520r/min
Mean pressure of variable pump inlet：1.1Mpa
Engine oil pressure：0.5Mpa
Mean pressure of steering gear：19.3Mpa
Engine coolant temperature：5℃
Mean pressure of priority valve：23.7Mpa
Engine oil temperature：138℃
Mean pressure of bucket cylinder：23.6Mpa
Engine fuel level:27L
Mean boom pressure of main valve：23.1Mpa
Transmission oil temperature：83℃
Mean pressure of main valve LS port：22.7Mpa
Transmission oil pressure：1.72Mpa
Mean pressure of foot valve：6.3Mpa
Filter blocking alarm：0
Mean pressure of brake pump inlet：0.4Mpa
Movable arm lifting alarm：0
Mean pressure of charging valve：15.8Mpa
Hydraulic oil temperature：73℃
Mean pressure of return inlet：0.3Mpa

Fault causes: 1.Thermostat occurs fault
2. Engine sysytem has fault

Fault locations: 1.Thermostat
2.Engine system

Maintenance methods: Replace thermostat

Figure 13. RBR loaders fault diagnosis results.

8. Conclusions and Future Work

This paper proposes an ontology-based method for the fault diagnosis of loaders. In this method, an ontology-based fault diagnosis model was introduced to achieve the integrating, sharing and reusing of fault diagnosis knowledge for loaders. Through the integration of CBR and RBR, a universal approach for effective and accurate fault diagnosis of all loaders was realized, with diagnostic results that can be displayed to users more directly in the developed application program. In addition, the program was validated through analyzing a case study.

In particular, in this highly expansible ontology model, the classes, properties and individuals in the fault diagnosis ontology model can be further updated and enhanced. Moreover, the fault diagnosis case base and fault diagnosis rule base are able to be improved and increased continually. Therefore, the fault diagnosis method presented in this paper is highly expansible, which gives it great potential to be applied to all construction machineries and even, hopefully, extended to decision analysis, automatic disassembly, intelligent early warning and other fields.

However, there are still some drawbacks in this method which need to be further strengthened in the future.

(1) The ontology model represented in this paper is relatively simple because the study of loaders fault diagnosis is not very comprehensive. In the future, more classes, object properties, data properties and individuals need to be designed to reflect the complex relationship between fault mode, fault causes and fault effects.

(2) By expanding the rule base and case base by defining more detailed rules and accumulating more cases, more fault diagnosis results (such as recommendation of fault maintenance shops, severity of fault harmfulness, fault alarming level, etc.) will be deduced.

(3) The values w_j and m_j will be calculated reasonably and accurately based on mathematical methods (such as analytic hierarchy process, expert judgment and fuzzy analysis, etc.) in the future.

(4) In the next step, more actual faults will be accumulated, and several indexes of fault diagnosis performance will be constructed to evaluate the accuracy of the method.

(5) The proposed fault diagnosis method is invalid when the data acquisition device of loaders cannot collect or loses running parameters. Therefore, the probability model of fault locations and fault causes of loaders will be evaluated and diagnosed based on historical operating parameters and the prediction algorithm in the future.

(6) Relative to the correction maintenance mode used in this paper, predictive fault diagnosis and maintenance is more important, because it can repair the faults in advance, so as to avoid greater damage to the loaders caused by faults. To realize predictive maintenance, firstly, auto-regressive and a moving average model and an artificial neural network need to be introduced, to predict operating parameters based on historical operating parameters. Secondly, based on predictive operating parameters, the fault diagnosis method combining ontology, CBR and RBR will still be executed, as in this paper. Finally, predictive fault causes, fault locations and maintenance measures can be achieved.

(7) Although we validated the approach to some extent, it really needs to be empirically evaluated as well. Hence, in the future, we will invite experts and maintenance shop professionals in the loaders fault diagnosis domain to diagnose faults empirically. By combining the approach results and their empirical results, the approach proposed in this paper can be further validated.

Acknowledgments: This paper is supported by the Nature Science Fund of China (NSFC) under Grant No. 51405187, National High Technology Research and Development Program 863 under Grant No. 2007AA04Z208, and Industry-University-Research Collaboration Project "Development of FW50GL wheel loader" under Grant No: FW/RD201640.

Author Contributions: Feixiang Xu and Wei Chen conceived and designed the proposed CBR and RBR based fault diagnosis methods; Feixiang Xu built the fault diagnosis ontology model. Bingwei Cao and Wei Chen collected fault diagnosis experiences and constructed fault diagnosis case base. Feixiang Xu and Chen Zhou designed fault diagnosis rule base. Bingwei Cao and Chen Zhou conducted experiments and analyzed the collected data; Feixiang Xu and Wei Chen developed the application program; Feixiang Xu wrote the paper. Xinhui Liu and

Chen Zhou revised the manuscript; Feixiang Xu, Xinhui Liu, Wei Chen, Bingwei Cao and Chen Zhou approved the final version.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhang, Z.; Shao, X.; Yu, D. Fault Diagnosis of a Wheel Loader by Artificial Neural Networks and Fuzzy Logic. In Proceedings of the IEEE International Conference on Robotics, Automation and Mechatronics, Bangkok, Thailand, 1–3 June 2006; pp. 1–5.
2. Li, Y.; Feng, X.; Yang, L. Intelligent Fault Diagnosis Methodology of Hydraulic Traveling System for Construction Machinery. In Proceedings of the IEEE International Conference on Industrial Informatics, Beijing, China, 25–27 July 2012; pp. 139–144.
3. Chadli, M.; Abdo, A.; Ding, S.X. H_/H∞ Fault Detection Filter Design for Discrete-time Takagi-Sugeno Fuzzy System. *Automatica* **2013**, *49*, 1996–2005. [CrossRef]
4. Gao, Z.; Cecati, C.; Ding, S.X. A Survey of Fault Diagnosis and Fault-Tolerant Techniques—Part I: Fault Diagnosis with Model-Based and Signal-Based Approaches. *IEEE Trans. Ind. Electron.* **2015**, *62*, 3757–3767. [CrossRef]
5. Ding, S.X.; Yang, Y.; Zhang, Y.; Li, L. Data-driven realizations of kernel and image representations and their application to fault detection and control system design. *Automatica* **2014**, *50*, 2615–2623. [CrossRef]
6. Youssef, T.; Chadli, M.; Karimi, H.R.; Wang, R. Actuator and sensor faults estimation based on proportional integral observer for TS fuzzy model. *J. Frankl. Inst.* **2017**, *354*, 2524–2542. [CrossRef]
7. Shi, P.; Zhang, Y.; Chadli, M.; Agarwal, R.K. Mixed H-infinity and passive filtering for discrete fuzzy neural networks with stochastic jumps and time delays. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 903–909. [CrossRef] [PubMed]
8. Chibani, A.; Chadli, M.; Shi, P.; Braiek, N.B. Fuzzy Fault Detection Filter Design for T-S Fuzzy Systems in Finite Frequency Domain. *IEEE Trans. Fuzzy Syst.* **2017**, *25*, 1051–1061. [CrossRef]
9. Gao, Z.; Cecati, C.; Ding, S.X. A Survey of Fault Diagnosis and Fault-Tolerant Techniques—Part II: Fault Diagnosis with Knowledge-Based and Hybrid/Active Approaches. *IEEE Trans. Ind. Electron.* **2015**, *62*, 3768–3774. [CrossRef]
10. Yang, B.; Zheng, W.; Ke, X. Forecasting of Industrial Water Demand Using Case-Based Reasoning—A Case Study in Zhangye City, China. *Water* **2017**, *9*, 626.
11. Ketler, K. Case-Based Reasoning: An Introduction. *Expert Syst. Appl.* **1993**, *6*, 3–8. [CrossRef]
12. Bryant, S.M. A Case-Based Reasoning Approach to Bankruptcy Prediction Modeling. *Intell. Syst. Account. Financ. Manag.* **2015**, *6*, 195–214. [CrossRef]
13. Elhdad, R.; Chilamkurti, N.; Torabi, T. An Ontology-based Framework for Process Monitoring and Maintenance in Petroleum Plant. *J. Loss Prev. Process Ind.* **2013**, *26*, 104–116. [CrossRef]
14. Verstichel, S.; Ongenae, F.; Loeve, L.; Vermeulen, F.; Dings, P.; Dhoedt, B.; Dhaene, T.; Turck, F.D. Efficient Data Integration in the Railway Domain through an Ontology-based Methodology. *Transp. Res. Part C* **2011**, *19*, 617–643. [CrossRef]
15. Chira, O.; Chira, C.; Roche, T.; Tormey, D.; Brennan, A. An Agent-based Approach to Knowledge Management in Distributed Design. *J. Intell. Manuf.* **2006**, *17*, 737–750. [CrossRef]
16. Panetto, H; Dassisti, M; Tursi, A. ONTO-PDM: Product-driven Ontology for Product Data Management Interoperability within Manufacturing Process Environment. *Adv. Eng. Inform.* **2012**, *26*, 334–348. [CrossRef]
17. Yang, D.; Miao, R.; Wu, H.; Zhou, Y. Product Configuration Knowledge Modeling Using Ontology Web Language. *Expert Syst. Appl.* **2009**, *36*, 4399–4411. [CrossRef]
18. Yang, D.; Dong, M.; Miao, R. Development of a Product Configuration System with an Ontology-based Approach. *Comput.-Aided Des.* **2008**, *40*, 863–878. [CrossRef]
19. McGuinness, D.L.; Van Harmelen, F. OWL Web Ontology Language Overview; W3C Recommendation. Available online: http://www.w3.org/TR/owl-features/ (accessed on 2 August 2014).
20. Shekarpour, S.; Auer, S.; Ngonga Ngomo, A.C.; Gerber, D.; Hellmann, S.; Stadler, C. Generating SPARQL Queries Using Templates. *Web Intell. Agent Syst.* **2013**, *11*, 283–295.
21. Huang, X.; Yi, J.; Zhu, X.; Chen, S. A Semantic Approach with Decision Support for Safety Service in Smart Home Management. *Sensors* **2016**, *16*, 1224. [CrossRef] [PubMed]

22. Wang, Y.; Yi, J.; Zhu, X.; Luo, J.; Ji, B. Developing an Ontology-Based Cold Chain Logistics Monitoring and Decision System. *J. Sens.* **2015**, *2015*, 1–8. [CrossRef]
23. Huang, X.; Yi, J.; Chen, S.; Zhu, X. A Wireless Sensor Network-based Approach with Decision Support for Monitoring Lake Water Quality. *Sensors* **2015**, *15*, 29273–29296. [CrossRef] [PubMed]
24. Zhu, X.; Yi, J.; Huang, X.; Chen, S. An Ontology-based Knowledge Modelling Approach for River Water Quality Monitoring and Assessment. In Proceedings of the 20th International Conference on Knowledge Based and Intelligent Information and Engineering Systems, York, UK, 5–7 September 2016; pp. 335–344.
25. Chen, R.; Zhou, Z.; Liu, Q.; Pham, D.T.; Zhao, Y.; Yan, J.; Wei, Q. Knowledge modeling of fault diagnosis for rotating machinery based on ontology, Industrial Informatics (INDIN). In Proceedings of the 2015 IEEE 13th International Conference on Industrial Informatics, Cambridge, UK, 22–24 July 2015.
26. Peng, H.; Chang, D.; Wang, Y. Fault Diagnosis of Conveyor Based on Ontology. *Sens. Transducers* **2013**, *157*, 338–345.
27. Medina-Oliva, G.; Voisin, A.; Monnin, M.; Leger, J.B. Predictive diagnosis based on a fleet-wide ontology approach. *Knowl.-Based Syst.* **2014**, *68*, 40–57. [CrossRef]
28. Castro, A.; Sedano, A.A.; García, F.J.; Villoslada, E.; Villagrá, V.A. Application of a Multimedia Service and Resource Management Architecture for Fault Diagnosis. *Sensors* **2018**, *18*, 68. [CrossRef] [PubMed]
29. Samirmi, F.D.; Tang, W.; Wu, Q. Fuzzy Ontology Reasoning for Power Transformer Fault Diagnosis. *Adv. Electr. Comput. Eng.* **2015**, *15*, 107–114. [CrossRef]
30. Ma, G.; Jiang, L.; Xu, G.; Zheng, J. A Model of Intelligent Fault Diagnosis of Power Equipment Based on CBR. *Math. Probl. Eng.* **2015**, *2015*, 1–9. [CrossRef]
31. Olsson, T.; Gillblad, D.; Funk, P.; Xiong, N. Explaining Probabilistic Fault Diagnosis and Classification Using Case-based Reasoning. In Proceedings of the International Conference on Case-Based Reasoning, Cork, Ireland, 29 September–1 October 2014; pp. 360–374.
32. Dou, D.; Yang, J.; Liu, J.; Zhao, Y. A Rule-based Intelligent Method for Fault Diagnosis of Rotating Machinery. *Knowl.-Based Syst.* **2012**, *36*, 1–8. [CrossRef]
33. Deng, X.-W.; Gao, Q.-S.; Zhang, C.; Hu, D.; Yang, T. Rule-based Fault Diagnosis Expert System for Wind Turbine. In Proceedings of the ITM Web of Conferences, 26 May 2017; pp. 1–9.
34. Olsson, E.; Funk, P.; Bengtsson, M. Fault Diagnosis of Industrial Robots Using Acoustic Signals and Case-Based Reasoning. In Proceedings of the European Conference on Case-Based Reasoning, Madrid, Spain, 30 August–2 September 2004; pp. 686–701.
35. Vo, D.P.; Macchion, D. A Use of Case-based Reasoning Technique in Building Expert Systems. *Future Gener. Comput. Syst.* **1993**, *9*, 311–319. [CrossRef]
36. Guo, R.; Peng, Y.; Hu, J. Research on high creative application of case-based reasoning system on engineering design. *Comput. Ind.* **2013**, *64*, 90–103. [CrossRef]
37. Chen, S.; Yi, J.; Jiang, H.; Zhu, X. Ontology and CBR Based Automated Decision-making Method for the Disassembly of Electromechanical Products. *Adv. Eng. Inform.* **2016**, *30*, 564–584. [CrossRef]
38. Dendani-Hadiby, N.; Khadir, M.T. A Case based Reasoning System Based on Domain Ontology for Fault Diagnosis of Steam Turbines. *Int. J. Hybrid Inf. Technol.* **2012**, *3*, 89–103.
39. Dendani-Hadiby, N.; Khadir, M.T. A Fault Diagnosis Application Based on a Combination Case-based Reasoning and Ontology Approach. *Int. J. Knowl.-Based Intell. Eng. Syst.* **2013**, *17*, 305–317. [CrossRef]
40. Zhou, A.; Yu, D.; Zhang, W. A Research on Intelligent Fault Diagnosis of Wind Turbines Based on Ontology and FMECA. *Adv. Eng. Inform.* **2015**, *29*, 115–225. [CrossRef]
41. Huang, X.; Yin, L. An Ontology Based Approach with Intelligent Decision for Off-Line Diagnosis of ECU. *Int. J. Simul. Syst. Sci. Technol.* **2016**, *17*, 1–8.
42. Tung, Y.H.; Tseng, S.S.; Weng, J.F.; Lee, T.P.; Liao, A.Y.; Tsai, W.N. A Rule-based CBR Approach for Expert Finding and Problem Diagnosis. *Expert Syst. Appl.* **2010**, *37*, 2427–2438. [CrossRef]
43. Zhou, Q.; Yan, P.; Xin, Y. Research on a Knowledge Modeling Methodology for Fault Diagnosis of Machine Tools Based on Formal Semantics. *Adv. Eng. Inform.* **2017**, *32*, 92–112. [CrossRef]
44. Wang, D.; Tang, W.H.; Wu, Q.H. Ontology-based Fault Diagnosis for Power Transformers. *Power Energy Soc. Gen. Meet.* **2011**, *45*, 1–8.
45. Wu, C.-Y. *Expert System for Tractor Fault Diagnosis System Based on Ontology*; South China Agricultural University: Guangzhou, China, 2012.

46. Ion, S.; Bogdan, I. Source Code Plagiarism Detection Method Using Protégé Built Ontologies. *Inform. Econ. J.* **2013**, *17*, 75–86.
47. Alderfer, C.P. Clarfing the Meaning of Mentor-Protégé Relationships. *Consult. Psychol. J. Pract. Res.* **2014**, *66*, 6–19. [CrossRef]
48. Uchibayashi, T.; Apduhan, B.O.; Shiratori, N. A domain specific sub-ontology derivation end-user tool for the Semantic Grid. *Telecommun. Syst.* **2014**, *55*, 125–135. [CrossRef]
49. Hu, Q.; Wang, H.; Xu, G.; Wu, D.; Yao, Y.; Lu, Q. The IDEF5-based Modeling of Process Ontology for Rapid Production Enterprise. *Mach. Des. Manuf. Eng.* **2008**, *37*, 6–10.
50. Zhou, Y. *Research on Machine Learning Based on Ontology*; Shenyang University of Technology: Shenyang, China, 2007.
51. Chaware, S.; Rao, S. Integrated Approach to Ontology Development Methodology with Case Study. *Int. J. Database Manag. Syst.* **2010**, *2*, 486.
52. Scheuermann, A.; Leukel, J. Supply Chain Management Ontology from an Ontology Engineering Perspective. *Comput. Ind.* **2014**, *65*, 913–923. [CrossRef]
53. Schank, R.C. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*; Cambridge University Press: Cambridge, UK, 1983.
54. Wang, S.L.; Hsu, S.H. A Web-based CBR Knowledge Management System for PC Troubleshooting. *Int. J. Adv. Manuf. Technol.* **2004**, *23*, 523–540. [CrossRef]
55. Relich, M.; Pawlewski, P. A case-based reasoning approach to cost estimation of new product development. *Neurocomputing* **2018**, *272*, 40–45. [CrossRef]
56. Derbinsky, N.; Laird, J.E. Effective and Efficient Forgetting of Learned Knowledge in Soar's Working and Procedural Memories. *Cogn. Syst. Res.* **2013**, *24*, 104–113. [CrossRef]
57. Wang, X.; Wong, T.N.; Fan, Z.-P. Ontology-based supply chain decision support for steel manufacturers in China. *Expert Syst. Appl.* **2013**, *40*, 7519–7533. [CrossRef]