

The ontology-based approach to data storage systems technical diagnostics

*Olga V. Mamoutova*¹, *Svetlana V. Shirokova*¹, *Mikhail B. Uspenskij*¹, *Aleksandra V. Loginova*^{1,*}

¹Peter the Great St. Petersburg Polytechnic University, 195251, Polytechnicheskaya Street, 29, St. Petersburg, Russia

Abstract. Monitoring and diagnosing the state of data storage systems, as well as assessing reliability and troubleshooting, require a formalized health model. A comparative analysis of existing knowledge representation methods has shown that an ontological approach is well suited for this task. This paper introduces a machine-represented data storage reliability ontology with an expert health model as baseline data. Classes of the ontology include the key terms of the reliability domain. Stated requirements for data interpretation tools allow further processing of the ontology-based knowledge base. Described ontology-based diagnostic systems have shown their applicability in the case of data storage systems in the construction industry.

1 Problems of data storage system diagnostics

A data storage system integrates multiple hardware and software components to store, manage and protect user data. A typical hardware platform consists of storage controllers, cache memory, chassis, and drives. Software of a platform implements a data path for the most critical data processing, a control path to manage a system configuration and service path for non-critical, but necessary system monitoring.

A fault occurring in any part of the system may lead to malfunction of a component and eventually cause loss or corruption of information. To prevent a system failure or diminish its consequences a data storage system should implement technical diagnostics as a part of a service path. This diagnostic service should monitor the state of hardware and software components, analyze any change of health parameters values, detect occurring faults, promptly notify the operator of changes in the system's performance and suggest recommendations for recovery.

Design of an effective diagnostic solution requires the following information:

1. What set of test parameters will show the consequences of all expected system faults in a noncontradictory manner.
2. How frequently the values of test parameters should be polled for adequate technical state inspection and prediction. Also, which test parameters should be polled with varying frequencies.

* Corresponding author: alexandra-lo@yandex.ru

3. How the designer could verify the algorithm of technical diagnosis and diagnosis equipment.

To answer these questions a designer has to create a diagnostic model that defines the levels of system performance degradation and expected failure modes. Thus an adequate diagnostic model must satisfy the following requirements:

- the failure modes must be representative of the system and its environment;
- system states in the model must be mapped to the failure modes;
- the levels of system performance degradation must be mapped to the defined system states;
- any defined system state must be in one-to-one correspondence with a particular set of test parameters values.

In addition a good diagnostic model should allow:

- expansion with new failure modes in case of new results of failure modes and effects analysis;
- configuration, when the parameters of the system change;
- variability in how detailed different components of the system are represented.

In general a manufacturer of a data storage system has an expert knowledge regarding the system dependability and quality of service – an expert health model. Usually this knowledge is not very well structured and doesn't have a formal, but rather verbal representation. For example, an expert health model can define the following levels of system performance: fully operable, vulnerable and critical. In this case a fault of CPU in storage controller puts the system into a vulnerable state and the loss of all controllers puts the system into a critical state.

This kind of expert health model would describe a set of known failure modes based on life data and user feedback. However, it does not show the connection between test parameters, particular changes of their values (symptoms) and known failure modes. It does not show the dependency between failure modes, system states and system configuration as well. And the main drawback of an expert health model is inability to use it for computer-aided automated diagnosis.

As a solution we present a formal approach to technical diagnosis of data storage systems using knowledge base methods.

2 Overview of formalisms for knowledge representation

There are several commonly used knowledge representation models: semantic networks, frame models, production models and logical models.

The semantic network is an information model of the domain in the form of a directed graph, which vertices correspond to the objects of the domain, and the arcs (edges) define the relations between them [1-2].

The frame model is a technological model of a person's memory and consciousness that is systematized as a unified theory [1-4].

A production model is a rule-based model that allows you to represent knowledge in the form of sentences like "If (conditions), then (action)" and is similar to a condition in programming languages if ... else ... [1-3].

Logical models are used to represent knowledge, expressed in the form of statements, axioms, which are characterized by the accuracy of determining the meaning of an expression, the possibility of building a knowledge base of a modular type on their basis, and the compactness of written expressions [1-2].

When using these models, there is a great risk of losing a significant part of the knowledge. The main idea of creating a model of this type is to consider all the information that is used to find the optimal plan of tasks, and a set of facts and statements presented in

the form of formulas and equations. Comparison of various models of knowledge representation showed that none of the approaches satisfy all of the requirements stated before.

3 Ontology of Data Storage System Dependability

We propose to implement an ontological approach since ontology is a model that provides an adequate representation of knowledge and is devoid of the drawbacks of the approaches listed above.

Ontology is a formal, exact specification of shared conceptualization [5]. This is a formal representation of a domain using a conceptual framework that defines the terminology of the subject area. The machine representation of the ontology allows to automate the processing of the data presented in it: basic concepts and concepts, as well as links between them [6]. For example, J. Zhou et al. use an ontological approach to model software reliability [7].

Ontology is required to link the concepts of faults, errors, error classes and test parameters. Some existing ontologies include the concepts of reliability of technical systems (e.g. ACM CS Classification). Existing ontologies are of a general nature and do not solve the problem posed though. In addition, the existing ontologies are presented in English and do not include Russian terms.

Therefore, it is necessary to develop a new ontology with a technical diagnostics as a domain, a data storage condition monitoring and diagnostics as a scope, and a developer of the diagnostic system as a target user of the ontology. Requirements for the ontology are given in Table 1.

Table 1. Requirements for the Ontology.

Competence	Ontology Classes	Diagnostic Task	Explanation
know how to interpret informational signs	Symptom Error Systemstate	control of technical condition	A separate set of observed informational signs can be associated with a subset of errors characterizing the state of the system. The health symptom of the storage system can be defined as the worst among all the signs corresponding to these errors.
know which informational signs you need to track and in what order	Error Symptom	control of technical condition	It can be implemented by ranking informational signs, for example, on primary (associated with a large number of types of errors) and confirmatory (associated with a smaller number or only with one type of errors).
be able to fail	Symptom Fault Error	verification of diagnostic algorithms	Depending on the level of abstraction, which requires the implementation of the introduction of faults, a specific fault can be simulated as a change in the internal state of the system or as a change in information signs.
know what errors and malfunctions can occur in the system	Risk factor Fault Error	technical prediction	Based on the data on the intensity of the flow of faults and taking into account the presence, absence or change of individual risk factors, it is possible to estimate the probability of errors.

know where to get information about fault rates	Manufacturer report	technical prediction	Background information on the probability of drive failures can be obtained from the official documentation of the manufacturer.
know how to prevent a malfunction or fix an error	Error Fault Reliability measure	decision making in case of changes in the technical condition	When solving the problem of controlling the technical condition: the decision-making subsystem should take measures to correct the detected errors. When solving the problem of predicting the technical condition: the decision-making subsystem should take preventive measures to change / implement means of preventing malfunctions.

Today there are two standards for ontology descriptions: OWL (Web Ontology Language) and RDF. At the same time, there are various tools to describe ontologies: define classes and relations in ontology, as well as build a knowledge base on their basis [8-9]. For the purposes of the current project we have set the following requirements for the development environment: support for one of the standard ontology description languages, the ability to export data, a graphical interface, open source code and support for plug-ins. All these requirements are met by the Protégé software [10], which is most often used for similar tasks in scientific and academic projects.

Based on the task, classes defined in the ontology correspond to the concepts specified in the expert health model and are supplemented with terms from the reliability domain (risk, failure, reliability improvement method) (Figure 1).

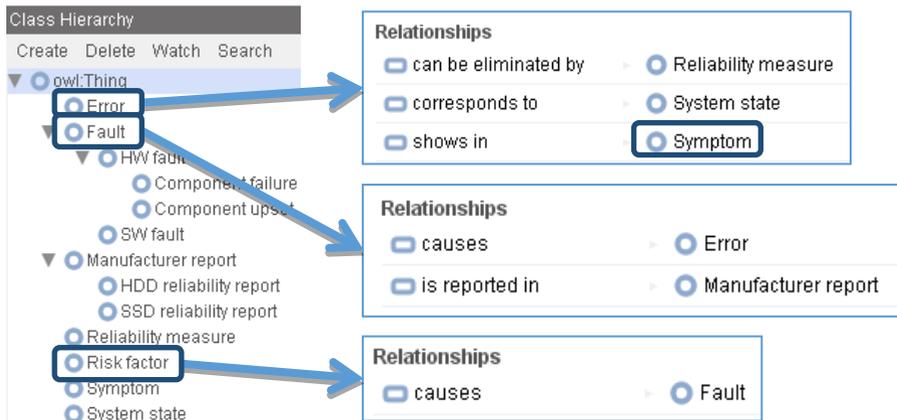


Fig. 1. Classes in the data storage health ontology.

4 Formal approach to health model

A way to formally represent conceptual relations is to define them on the domain space. For the purpose of this paper let us consider the following notions in the domain space: faults as the cause of system failure, errors as a characteristic of system failure, and symptoms that allow identification of a failure.

Let $F = \{f_1, f_2, \dots, f_n\}$ be a set of expected system faults, as described in expert health model and derived from the known failure modes, where each f_i can be defined with a

corresponding stochastic process. Let $S = \{s_1, s_2, \dots, s_m\}$ be a set of possible errors caused by system fault. And let $\Delta = \{\delta_1, \delta_2, \dots, \delta_p\}$ be a set of symptoms, that can be monitored via test parameters.

Now we can say, that an error s_k is in a direct consequence relation ρ_1 with a fault f_i , if an error s_k occurs only after the fault f_i occurs. The same way a symptom δ_j is in a direct consequence relation ρ_2 with an error s_k , if the error s_k always shows as the symptom δ_j .

We shall call the relation (f_i, δ_j) as a consequence relation φ , when there is an error s_k that holds direct consequence relations (f_i, s_k) и (s_k, δ_j) with f_i and δ_j .

For example, in a data storage system the following are in the consequence relations: f_i – a CPU upset in a storage controller, s_k – loss of the controller, δ_j – absence of messages from the controller in system logs. Another example: f_i – a hard drive sector damage, s_k – partial data loss, δ_j – increment of SMART reallocated sector counter or δ_j – increase of a corrected error count (if error correcting codes are present).

Thus a formal health model $\langle F, \Delta, \varphi \rangle$ would define all the pairs (f_i, δ_i) , that hold consequence relations φ with each other. Such a model can be used as a diagnostic model (Fig. 2). When annotated with functional relations, the formal model becomes a mathematical reliability model, which represents system properties and system’s state transition dynamic.

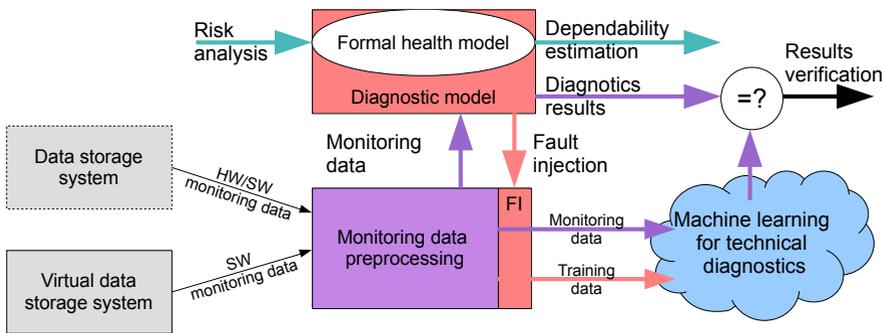


Fig. 2. Test system with a diagnostic model based on a formal health model.

We set the following modeling goals:

1. Support fault injection via modification of the structured monitoring data.
2. Evaluate System reliability based on estimates of the probability of fault occurrence in its elements.
3. Define the set of test parameters to be polled and diagnose a data storage failure based on system monitoring data.
4. Determine the necessary frequency of data collection and a subset of the analyzed test parameters as a part of the complex examination methodology.
5. Verification of machine learning training and testing results.
6. Validation of the monitoring data in order to exclude incorrect data from subsequent analysis.

The corresponding model input should be the following:

- fault intensity parameters based on a risk assessment or as the output of a fault injection module;
- structured monitoring data with the composition according to the architecture of a specific storage system.

And the corresponding model output should be the following:

- evaluation of the system reliability;
- system performance diagnosis;
- structured monitoring data, modified in accordance with the fault injection methodology.

Additional model requirements are the following:

- rapid reliability evaluation at the early design stages and during the operation of the system;
- flexibility and scalability in order to customize the model in accordance with the data storage configuration parameters;
- the possibility of expanding the model with a detailed representation of system elements.

5 Discussion

One of the ways to use the ontology is to develop algorithms for condition monitoring and diagnostics in a data storage system. As an example, such an algorithm would determine the current state of the system by searching for the closest match in a set of system states based on comparison of the monitoring data and the symptoms of the corresponding states, all described in the ontology. As a result of the comparison, the state of the system is recognized as a class of the existing ontology, the distance to which from the actual class of the ontology is minimal, after which it would determine whether this state is a warning, an error or a critical failure.

A basic methodology of creating a knowledge base on the basis of a specific ontology includes a non-automated transformation of an expert health model. However, to build a complex knowledge base and provide further data processing, it is necessary to develop a set of tools, including automation tools for creating the knowledge base and knowledge output tools to support the diagnostic algorithm. Moreover, in order to ensure that an ontology and knowledge base can be used, it is necessary to perform an ontology assessment and verification: to verify the ontology's correctness and consistency, as well as to verify that the ontology is consistent with its purpose.

One way to build a knowledge base includes the following steps:

1. Create instances for the following classes: Fault, Error, System state, Symptom (the core of the ontology). It is advisable to perform this step for each class in parallel and independently, so as not to miss possible errors in the source data (as identified in step 4).
2. Set relationships between instances of classes.
3. Check the knowledge base for the absence of instances of classes: whether it was possible to specify all the links in the ontology with instances of the listed classes.
4. If an isolated instance of a class is found, analyze the cause of this error and supplement the knowledge base accordingly. Save a record of this error. Return to step 2.
5. If it is necessary to solve a particular diagnostic problem, supplement the knowledge base with instances of one of the corresponding classes: Risk factor, Manufacturer report, Reliability measure.

Options of how to build a formal health model include the following:

1. A fault tree with an analytical solution based on a set of events and relationships between events described in the ontology.
2. A Petri net or one of the variants of extended Petri nets (for example, the Stochastic Activity Network) based on a variety of events and relationships between events described in the ontology.

To implement these two methods, it is required to explicitly determine functional dependencies to describe the cause-and-effect relationships between the system states in the knowledge base.

3. A finite state machine A with the transition graph, where vertices represent the changes in the internal states of the system s_k . Such a finite state machine can be created by system decomposition followed by functional analysis of the system, or by process mining based on the set of the monitoring data as the event logs. In this case, the data storage diagnostic can be considered as an state machine diagnostic task: for a given state machine A with the transition table it is required to find the current state s_k based on the values of the automaton outputs represented by the (not always fully defined) vector of output values $(\delta_1, \delta_2, \dots, \delta_p)$.

6 Approbation

These systems have shown their applicability by the example of data storage systems in the construction industry.

Ontology-based approach to data storage systems technical diagnostics was tested for a storage system in a construction company. Development of technical documentation and drawings is carried out using special software [11-14]. These application software products allow you to form and develop architectural solutions and their components.

During the study and analysis of individual structural elements, a set of models is formed. This set is a knowledge base that is stored in the data acquisition system. To improve the reliability of storing this information and diagnosing its storage system, an approach has been applied. A prototype of an expert system has been developed using a combined presentation model with semantic network and frame model as the basis.

7 Acknowledgement

The research is carried out with the financial support of the Ministry of Science and Higher Education of Russian Federation within the framework of the Federal Program "Research and Development in Priority Areas for the Development of the Russian Science and Technology Complex for 2014-2020". The unique identifier is RFMEFI58117X0023.

References

1. V.V. Goryunova, *Logical basis of knowledge representation in intelligent information systems* (Penza State University of architecture and construction, Penza, 2004)
2. O.A. Shabalina, A.V. Matokhina, *Systems of knowledge management in organizational and economic, production and technical systems. Models and languages of knowledge representation, Textbook* (Volgograd State Technical University, Volgograd, 2015)
3. P.N. Bibilo, V.I. Romanov, *Logical design of discrete devices using production-frame model of knowledge representation. Textbook* (Belarusian science, Minsk, 2011)
4. V.P. Deryabkin, *40 years of the Information systems and technologies Department of the Samara State Aerospace University* (Samara State Aerospace University, Samara, 2012)
5. L.S. Bolotova, *Decision support Systems. Part 2* (Yurayt Publishing House, Moscow, 2017)
6. N.F. Noy, D. Mcguinness, Knowledge Systems Laboratory **32**, 1-17 (2001)
7. J. Zhou, O. Eila, E. Antti, *Proc. of Software and Service Variability Management – Concepts, Models and Tools* (Helsinki, Finland, 2007)

8. A.J. Duineveld, R. Stoter, M.R. Weiden, B. Kenepa, V.R. Benjamins, *Int. J. of Human-Computer Studies* **52**, 1111–1133 (2000)
9. Z.H. Malik, *Proc. of 2017 Computing Conference*, 576–584 (London, UK, 2017)
10. A. Musen, *AI Matters* **1(4)**, 4–12 (2015)
11. S.V. Shirokova, M.V. Bolsunovskaya, A.M. Gintciak, M.S. Zharova, *Proc. of the 3rd International Conference Ergo-2018: Human Factors in Complex Technical Systems and Environments* (ETU “LETI”, Saint Petersburg, 2018)
12. O. Novikova, S. Shirokova, A. Tabakova, A. Shadrin, *MATEC Web Conf.* **170**, 03006 (EDP Sciences, 2018)
13. A.A. Efremov, A.V. Loginova, B.D. Mikeladze, S.V. Shirokova, *Proc. of the 20th IEEE International Conference on Soft Computing and Measurements* (ETU “LETI”, Saint Petersburg, 2017)
14. V.N. Volkova, A.V. Loginova, S.V. Shirokova, E.A. Kozlovskaya, *Proc. of the 19th International Conference on Soft Computing and Measurements SCM 2016* (ETU “LETI”, Saint Petersburg, 2016)