A new method of monitoring performance of complex stored grain systems

Zhu Zesheng[1] and Sun Ling[2]

Abstract

Performance monitoring of complex stored grain systems, including dynamic collection, interpretation, and presentation of information, is very important for the stored grain management However, how to implement the monitoring used to make management decisions and perform the appropriate control actions on the systems has been being a very difficult problem In this paper, we discuss a method for performance monitoring of complex stored grain systems and how to use the monitoring as a management tool of the systems This method is used to determine the facilities needed to design and construct the system of monitoring performance of complex stored grain systems Some important problems related to the system implementation, including concepts, monitoring model, monitoring information processing, monitoring information dissemination and presentation, are deeply discussed From the discussion, the generic monitoring services are important tools for implementing stored grain management system and for debugging during the system development On the other hand, monitoring services are also essential for process control and management automation in management of complex stored grain systems. These monitoring services are based on a set of monitoring functions and may be used as a practical framework for the implementation of an advanced stored grain management system For showing practical application of the method, a typical system for the performance monitoring of complex stored grain system was implemented. The system has been used to monitor, analyze and process a number of data from many stored grain entities. The final experiment results show that the method performance is very satisfactory.

[1] Nanjing Naval Institute of Electronic Engineering, Nanjing 211800, P R China

[2] Corresponding Author, Jiangsu Academy of Agricultural Sciences, Nanjing 210014, P R China

Introduction

The management of complex stored grain system is very important for economic development of developing country. Especially, systems of stored grain have increased dramatically in both size and complexity in the last few years However, the new power brought with modern information processing technology creates greater vulnerability (Neumannm, 1992) Since fault is inevitable, quick detection, identification and recovery are crucial to make the systems more robust and their operation more reliable As systems of stored grain become more heterogeneous and more hardware and software from various vendors are used, the whole picture of the specification becomes bewildering This brings out the need for a unified approach or principles to the area of performance monitoring of complex stored grain system

Monitoring, the dynamic collection, interpretation, and presentation of information about complex stored grain system is needed for management of the stored grain system The information gathered is used to make management decisions and perform the appropriate control actions on the system. Management of complex stored grain system involves monitoring its activity, making management decisions and performing control actions to modify behavior Most of works on monitoring has concentrated on monitoring mechanisms related to simple stored grain system However, in order to automate the monitoring of complex stored system, it is necessary to be able to represent and manipulate monitoring policy within the system These objectives are typically set out in the form of general policies into a number of more specific policies to form a policy hierarchy in which each policy in the hierarchy represents, to its maker, his plans to meet his objectives and, to its subject, the objectives which must be plans to meet.

The task of the performance monitoring is to keep track of stored grain status, which include both severity and extent of possible disasters of stored grain, and trigger control actions when necessary The process can be divided into the monitoring process and the control process. The monitoring process involves collecting information about short-term or long-term behavior of stored grain and interpreting the

semantics of the collected information. The control process affects the state of dangerous stored grain according to the interpreted information to achieve a desired outcome Thus, there are two kinds of major issues in performance monitoring of complex stored grain system as the follows

(1) Monitoring information infrastructure: Any performance monitoring system must be constructed on top of the underlying monitoring information model on which the representation schemes and operations are based. Given that a disaster of stored grain occurs, several issues are confronted when designing the infrastructure of the performance monitoring information. They include the monitoring information representation and information distribution strategy

(2) Automatic and adaptive monitoring: The maintenance of a large number of objects related to stored grain disasters in monitoring information base (MIB) needs to be done automatically to keep the status information up-to-date Monitoring applications of stored grain disaster distribution can then easily identify and update objects This changes the distribution of disaster entities Either remedial or preventive monitoring schemes of performance of complex stored grain system need to be triggered automatically by the performance alarms, which again depend on automatic interpretation of performance measurement and influence This measurement interpretation implies that the monitoring system needs to keep track of the performance patterns and perform adaptive monitoring

There are a number of fundamental problems associated with monitoring of complex stored grain system For example, quality of stored grain means time that it can be stored in safety status However, it is very difficult to obtain a global, consistent view of all stored grain components related to stored grain quality in complex stored grain system through collecting event reports. Because estimation errors of real time temperature, moisture and humidity variables in the reporting of events may result in these events being processed in incorrect methods Another problem is that the monitoring system may itself compete for limited resources with the stored grain system being observed and so modify its behavior. To solve these problems, the monitoring system must provide a set of general functions for generating, processing, disseminating, and presenting monitoring information

Policy hierarchies

The generic characteristics of a performance monitoring policy can be represented as an object A number of operations can be performed on it The main motivation for understanding hierarchical relationships between policies is to determine what is required for the satisfaction of policies The general concept of policy hierarchies was discussed in reference (Moffett and Sloman, 1991) If a high-level policy is defined or changed, it should be possible to decide what lower-level policies must be created or changed. Another one is for analysis to see whether the set of lower-level policies actually fulfill the higher-level policy, by providing complete cover over all the target objects and actually meeting the policy goals Therefore, there is a need to be able to formally describe the relationships between high-level policies, refined lower-level policies, and the mechanisms, actions, and procedures which finally implement them. These policy hierarchies need to be constructed in such a way that a human manager of system of monitoring performance of complex stored grain system can determine whether the policies for the performance monitoring are being satisfied

Attributes

The attributes of the policy can be shown in Fig. 1. These attributes can be given as follows.

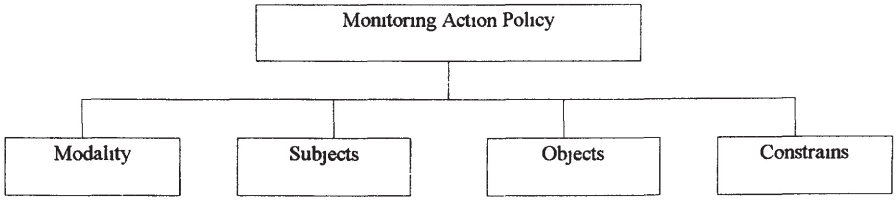

Fig. 1. Monitoring Action Policy Attributes

(1) Modality policies: They can be divided into two different classes of policies (a) Imperative policies: An agent has the imperative power to carry out an action (b) Authority policies: An agent has the legitimate power to carry out an action.

(2) Policy subjects: They are the entities to whom the policy is directed.

(3) Policy target objects: They are the objects at which the policy is directed.

(4) Policy objectives: They are expressed as a pair of

goals and target objects The policy goal defines either a high-level goal or a procedure Procedures are a defined sequence of actions

(5) Policy constraints: They are a set of predicates which constrain the applicability of the policy. They may be related to terms of general properties of stored grain disaster, such as temperature, moisture, pest and some other conditions.

Hierarchy

In the performance monitoring, a high-level policy may be used as the basis from which multiple lower-level policies are derived Various methods for refining the goals, partitioning the targets can be used for the derivation These methods can be discussed as the follows

(1) Partitioning targets: while the goal is the same, the target set of the lower-level policy may be a subset of the target set of the higher-level one. In order to ensure that the partitioning is complete, the whole target must be covered by lower-level policies

(2) Refining goal: The goal of a high-level may be refined into one or more lower-level goals, referring to the same target.

(3) Procedures: A policy may be refined by an unordered set of lower-level ones.

Action

There is a close relationship between making policies in a policy hierarchy and giving responsibility for a task of the performance monitoring The subject of a policy can be viewed as the entity responsible for carrying out the goals or actions defined by the policy. Thus, the specification of the monitoring responsibility can be modeled using the policy objects When a manager of performance monitoring system of complex stored grain system is assigned with responsibility for an objective, there is a need for an imperative policy to motivate the manager the power to perform the action. There are three different kinds of policy as the follows

(1) Action responsibility: It represents the sense of being 'responsible for' an objective

(2) Reporting responsibility: It can be achieved by the agent reporting to the manager on completion or at regular intervals

(3) Monitoring responsibility: There are two kinds of monitoring responsibility for complex stored grain system Subject monitoring is related to an action responsibility, where the action subject is the target of a monitoring policy Target monitoring occurs when targets of the action and monitoring policy are the same.

The accepting responsibility related to performance monitoring of complex stored grain system ensures that a policy subject accepts responsibility. There are two different kinds of accepting responsibility as follows

(1) Imperatives for automated managers: As soon as automated managers are made aware of a policy, they will begin to perform the actions required to satisfy the policy

(2) Imperatives for human managers: Their definition is similar to one of imperatives for automated managers.

Monitoring model

A good monitoring model is a framework for design and implementation of the monitoring system This model is used to determine the facilities needed to design and construct the system of monitoring complex stored grain system. Some important problems related to the model implementation include its concepts, terminology, and monitoring means. The object modeling technique (Rumbaugh, 1991) can be widely used for design and implementation of the monitoring model

Service and functions

The generic monitoring services built on the model are important tools for implementing management system of stored grain and for debugging it during the system development. On the other hand, monitoring services are also essential for process control and management automation in stored grain system, shown in Fig 2 Thus, such monitoring model is based on a set of monitoring functions and may be used as a practical framework for the implementation of management system of complex stored grain system. The main functions of monitoring model can be described as follows.

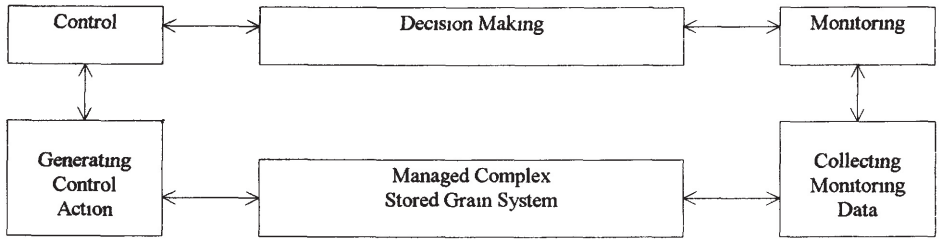

Fig.2. Management of Complex Stored Grain System

(1) Generation of monitored information, including status, events and trace reports
(2) Processing of monitored information, including to validate, combine and filter relevant information.
(3) Dissemination of required information to relevant objects who have subscribed to the service
(4) Presentation of monitored information to human users via flexible graphical facilities

Techniques and tasks

In the model, a managed object in complex stored grain system is defined as any stored grain, hardware or software component whose behavior can be controlled by a management system The object encapsulates its behavior behind an interface, which hides the internal details that are vital for monitoring purposes. Monitoring can be performed on an object or a group of related objects. The behavior of an object can be defined and observed in terms of its status and events Two kinds of monitoring techniques, time-driven monitoring and event-driven monitoring, are used respectively for acquiring periodic status information and information about the occurrence of events of interest. Furthermore, the monitoring model performs the following four monitoring tasks

(1) Generation: When some important events are detected, the related event and status reports are generated. These monitoring reports are used to construct monitoring traces, which represent current and historical views of activity of stored grain system
(2) Processing: It converts the raw monitoring data to the required format
(3) Dissemination: The monitoring reports are disseminated to the appropriate users, managers and processing agents
(4) Presentation: Gathered, processed, and formatted information is displayed to users

Systems and monitors

Various detection mechanisms can be used to identify the occurrence of events According to which mechanism is used, monitoring systems can be categorized into three types: hardware monitors, software monitors and hybrid monitors

Hardware monitors are separate objects that are used to detect events associated with an object or a group of objects The detection is performed by observation of system status or by using physical sensors or probes connected to a data acquisition system. Hardware monitors have been successfully used for monitoring various physical parameters of complex stored grain system, where a great deal of information is collected and processed rapidly.

On the other hand, software monitors usually makes use of simulation models to identify the occurrence of events. Software monitors present information in an application-oriented manner that is easy to understand and use, compared to the raw information generated by hardware monitors Software monitors can easily be replicated and are more flexible, portable, and easier to design and construct than hardware monitors The disadvantage of software monitors is that their simulation models require a great deal of input data and therefore interfere in both their application area and precision of monitored data For this reason, pure software monitors are not adequate for on-line, real-time monitoring of complex stored grain system.

Hybrid monitors are designed to employ the advantages of both hardware and software monitors while overcoming their disadvantages. Typical hybrid systems consist of an independent hardware device that receives monitoring information from typically stored grain components, and an independent software model that computes monitoring information for other similar stored grain components.

Alarm processing

Alarm correlation technique is widely used in the monitoring system. The conceptual approach to alarm correlation was discussed in (Aloni, et al , 1991) Interpretation and correlation of events has been analyzed in other areas such as electric power systems, nuclear-power-plant alarm management (Rellano, et al , 1991) and patient-care monitoring.

Alarm Correlation

The alarms are mediated by alarm messages about faults. A fault is an unsafe state in stored grain or a disorder occurring in the hardware or software of managed stored grain system. Faults happen within managed system components or stored grain, while alarms are external manifestations of faults. Alarms defined by designers and generated by stored grain and grain storage equipment are observable by stored grain managers.

Alarm correlation is a conceptual interpretation of multiple alarms so that new meanings are assigned to these alarms It is a generic process that underlies different management tasks of stored grain. Some typical operations relevant to alarm correlation are as follows

(1) Compression: the reduction of multiple occurrences of an alarm into a single alarm.
(2) Count: the substitution of a specified number of occurrences of alarms with a new alarm
(3) Suppression: inhibiting a low-priority alarm in the presence of a higher-priority alarm.
(4) Boolean: substitution of a set of alarms satisfying a Boolean pattern with a new alarm
(5) Generalization: reference to an alarm by its

superclass Alarm correlation may be used for fault isolation and diagnosis, selecting corrective actions, proactive maintenance, and trend analysis in complex stored grain system.

Conceptual framework

One of the major applications of alarm correlation is the fault diagnosis in complex stored grain system Not all faults exhibit alarms These faults can be recognized indirectly by correlating available alarms. Correlation between alarms due to a common fault is an equivalence relation

Alarm generalization is potentially very useful for management of stored grain It allows one to deviate from a microscopic perspective of management events of stored grain and view situations from a higher level There are two ways for alarm generalization. The first is subsumption of lower-level alarm classes by a higher-level class This generalization process may utilize alarm class/subclass hierarchies The second is interpretation of simultaneous events or events happening within a defined time interval as a qualitatively new complex situation

The conceptual framework of alarm correlation contains the structural and behavioral components. The relational approach to monitoring complex system (Snodgrass, 1988) can be used for design and implementation of the conceptual framework The structural component is the description of the managed stored grain system It contains two major parts, the configuration model of stored grain system and the element class hierarchy of stored grain system The configuration model describes the managed objects, the connectivity and containment relations between them The element class hierarchy describes the managed object types and the class/subclass relationships between the types

The behavioral component describes the dynamics of alarm correlation It contains three major components: the message class hierarchy, the correlation class hierarchy, and correlation rules. The managed object classes, message classes, correlation classes and correlation rules are organized into hierarchies. These hierarchies are related by 'producer/consumer' dependencies Managed objects are producers of alarm messages, messages produce correlation, and rules are consumers of all the above components.

Managed classes describe types of stored grain equipment. Managed objects are organized into a hierarchy using class/subclass relations The root of the hierarchy contains the most general information common to all managed objects. The next level of the hierarchy describes the basic managed object classes Each subclass inherits parameters, values, and attributes The managed object hierarchy is an abstraction of physical managed objects The hierarchy is specific to an application

Configuration model The configuration model of stored grain system is constructed from the instances of individual managed objects The instances describe the actual physical or logical components of the managed stored grain system

Message Class All alarm messages produced by a specific managed object are organized into a message class hierarchy using the class/subclass relation Introduction of message classes simplifies the decision-making process of stored grain management. Each message class in the hierarchy contains a message-parsing pattern and a translation schema, common to a subset of all messages that belong to this class

Correlation class. A correlation class is a generalized description of the state of stored grain system based on interpretation of events of stored grain system The conditions under which the correlation is asserted are described in the correlation rules. Each assertion creates an instance of a correlation class. A correlation class contains components, a message template and parameters The components may be managed objects, alarm messages, or other correlations. Correlation components are used to pass information from a correlation rule to the asserted correlation. Parameters provide information about a correlation to higher-level correlations

Correlation Rules Correlation rules recognize events and assert correlations Different correlation rules may assert the same type of correlation The conditional part of a rule is a Boolean pattern built upon primary terms and relations The primary terms are messages, managed objects, correlations, and tests The action part of the rule contains executable commands. The key components of alarm processing are shown in Fig 3

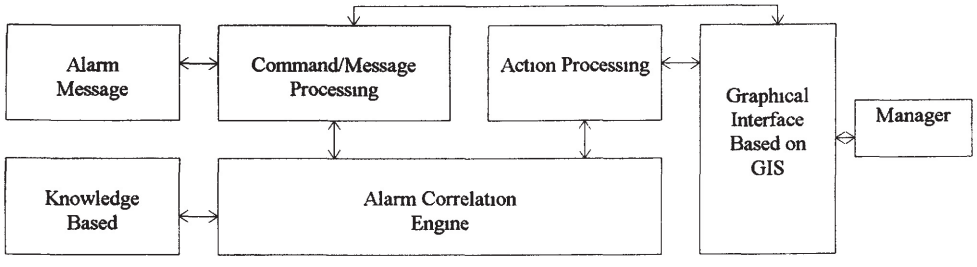

Fig. 3. Alarm Processing

Performance monitoring

From the policy hierarchy, the major components of the performance monitoring include monitoring information management, management of monitoring objects, and monitoring policy management As its name implies, monitoring information management is responsible for organization and recovery from the monitoring information and inflicted damages related to stored grain disaster Management of monitored objects is related to determination of object position and accommodation of the object distribution changes Monitoring policy management is responsible for providing the optimally monitoring policy for the performance monitoring by adjusting the performance monitoring decisions, and critical for efficient performance monitoring of complex stored grain system that is in a dynamic environment

One of the key elements of performance management is performance monitoring. Expert system can be used to support the performance management (Wagenbauer and Nejdl, 1993) In order to effectively manage a complex stored grain system, the administrator of the stored grain system must be able to determine when equipment and facilities within the stored grain system are operating in a degraded mode The administrator must also be able to determine where the stored grain is in a safe or unsafe state.

Basic model

Modeling information of the performance is to map distribution, characteristics, and events of the monitoring objects to objects in ED An inheritance hierarchy can represent a simple classification of performance object classes, where the elements class has three subclasses: distributions, characteristics and events Physical entities class has two subclasses: affected entities and geographic positions.

Performance data

Performance monitoring data in the policy hierarchy of the performance monitoring can be broadly classified into the followings

(1) Measurement data. The measurement data of complex stored grain system is the raw information that is received from the performance monitoring processes and various variables related to the monitoring The data provides the primary input for performance monitoring It represents the current status of the stored grain system Measurement data can be divided into two groups according to the general characteristics of policy of performance monitoring of stored grain system: persistent and perishable. The persistent data consists of measurement data, whose use is long-term, and therefore needs to be maintained permanently in the database On the other hand, perishable measurement data is of limited time use, so its current value is valid only until the stored grain system characteristic is being monitored

(2) Structural data In contrast to measurement data, structural data is composed of static performance information Unlike measurement data, structural data is valid even when the performance monitoring does not occur Most of structural data is stored at initiation time of the performance monitoring system

(3) Monitoring data Monitoring data captures the current selection of monitoring decision for performance The process of changing an existing set of monitoring decisions is usually completed by the performance managers of the policy hierarchy Alternatively, the changes may be automatically triggered as a function of the information in the measurement data In addition to the current settings of monitoring decisions, the monitoring database also stores a library of predefined monitoring decision settings that reflect the appropriate settings for a variety of common performance patterns

Thus, the performance monitoring systems based on policy hierarchy are responsible to monitor and interpret performance of complex stored grain

Policy management

The role of monitoring policy management of performance monitoring of complex stored grain is to manipulate the adjustable monitoring policy decisions in real time so that the performance can be efficiently monitored in order to reduce the stored grain loss Monitoring policy management from analysis for policy hierarchy is divided into two tasks as follows

(1) Monitoring policy evaluation that finds how changes in monitoring decisions reduce the loss of stored grain disaster; and

(2) Decision making on how to adjust the monitoring decisions The first task is essentially equivalent to finding a relationship between the quality of performance monitoring and the monitoring decisions, and may be required to estimate the quality and loss The second one is to decide what monitoring decision is selected for the performance monitoring.

Policy evaluation

The analytical techniques, such as probability theory, can be used for the monitoring policy evaluation However, they require unrealistic assumptions and tend to be mathematically untraceable as the structure of the evaluation

measure becomes complex. On the other hand, discrete-event simulation is a viable alternative to analytical techniques. Its major advantage is that it can be modeled with much less stringent assumptions, and more complex performance measures can be handled with relative ease. However, discrete-event simulation usually suffers from significant computational burden because a single simulation run represents only one realization of a stochastic process In order to obtain an accurate estimation of the performance monitoring quality under a given monitoring policy, several independent runs are needed, and these runs should be repeated.

Decision making

In the policy hierarchy, this task requires monitoring decision optimization, and can be accomplished by the learning and inference methods

Monitoring forms

Typically, performance management can be divided into monitoring and controling aspects. Performance monitoring refers to the collection of information on a state of complex stored grain system, while control refers to actions taken to improve performance of the stored grain system. The performance monitoring system must be flexible enough to satisfy current and future performance monitoring needs regardless of the technology or type of stored grain equipment being monitored The performance monitoring can take different forms as follows.

(1) One common use is hunting for a specific stored grain problem
(2) Another situation involves global monitoring The administrator will probably want to be advised of developing problems before they can affect the operation of stored grain system.
(3) A third possible use is the historical analysis of trends and stored grain quality It may be desirable to be able to perform statistical analysis of various performance data over some period of time.

Monitoring methods

The most elemental level of performance monitoring ultimately depends on measuring information about complex stored grain system. Measurements may take the form of counters. Counters are cumulative indicators that measure the occurrences of some event or the total number of units of some quantity. A managed object class of current data is a collection of counters and gauges that are used to monitor the state of stored grain system. There are different subclasses of current data Each subclass of current data corresponds to the type of resource being monitored. The subclasses define the particular set of counter and gauges that are applicable to a particular type of resource. Gauges give instantaneous measures of some quantity. Each type of resource being monitored may have a distinct set of counters and gauges that apply only to that type of resource

Reporting of Scheduled Statistics. A complete performance monitoring system deals with not only the collection and storage of performance data but also how the data is reported The system can simply use a command to read the attribute values of the relevant instances of current data, performance event record and history data.

Reporting of threshold Crossings. The performance data is saved at the end of the interval or reported at some scheduled point in time. In many cases, it is desirable to notify the administrator immediately when some event occurs Managed objects of threshold data contain the thresholds and hysteresis levels associated with the counters and gauges of current data If a counter or gauge in the current data passes a threshold, a notification is issued. The managed object instances of current data and managed object instances of threshold data are related by a relationship This relationship is used to allow different ways of using thresholds. Changing the threshold affects the monitoring of many managed object instances

Application

The information of performance monitoring can be used to complete performance management of stored grain system. The performance management functionality is broken down into different functional units (Balzer, et al , 1982) Functional units represent sets of capabilities and provide a way for the central system and stored grain element to inform each other of which capabilities they can support. Various functional units are supported by the model The only additional object classes required performance monitoring would be subclasses of current data, threshold data, and possibly an event forwarding discriminator

In this scenario, the managed object instances of current data will be created by the element. For each of the resource, such as warehouses being monitored, the element would instantiate an appropriate managed object of current data. The instance of current data would be contained within the managed object representing the resource The element selects the options such as the performance summary interval and the set of counters and gauges. The current data object instances would point to one or more threshold data objects that define the conditions causing the spontaneous performance reports to be issued. Principle of performance monitoring is shown in Fig 4.

Monitoring implementation

Our approach to solve the above problems of performance monitoring: information distribution strategy and automatic/adaptive monitoring are to incorporate learning and

inference abilities into performance monitoring system of complex stored grain system to automate the process of global view construction, measurement interpretation, problem forecasting, problem diagnosis and decision making related to the performance monitoring

Performance monitoring information in complex stored grain system describes the status and events associated with an object or a group of objects under scrutiny Such information can be represented by individual status and event reports

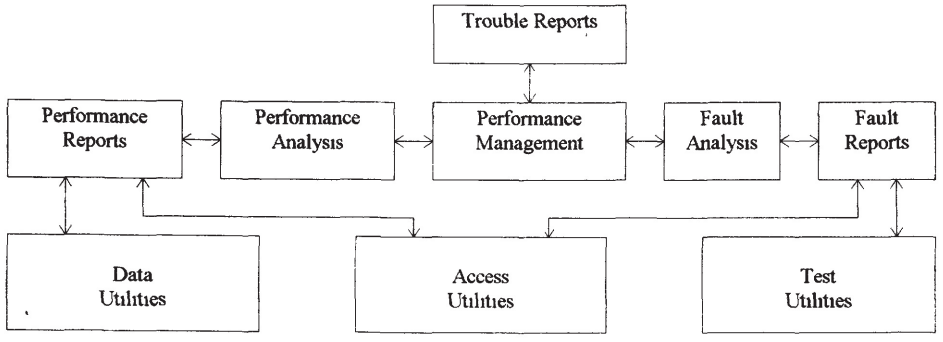

Fig. 4. Performance Monitoring

Global view

To build the information infrastructure about the performance monitoring, a set of global views is constructed with the help of the above policy hierarchy A global view is a virtual object class defined from MIB via logical rules. From the relationship between making policies, these global views serve as windows through which management applications can access physical entities about the performance monitoring

To implement the above automatic and adaptive abilities, the performance patterns related to policy attributes must be learned from a historical database which contains a chronological measurement trace. These discovered patterns have been represented and describe the correlation between objects of stored grain. Based on these performance patterns and prespecified domain knowledge, forward and backward inference can be triggered to access global views, predict performance status, fire control actions, and reported problems. Fig. 5 is an abstract decision process model of performance monitoring system of complex stored grain system. The extensional database (ED) is used to represent the basic facts about distribution, severity and performance measurement alarms, and events of stored grain disaster Each disaster has an associated ED which is its view about the disaster The historical database (HD) is the temporal historical database which encodes time in the disaster trace. Performance patterns are learned from HD and stored in Pattern knowledge Base (PKB) The domain knowledge base (DKB) is for prespecified problem solving and general relationship knowledge.

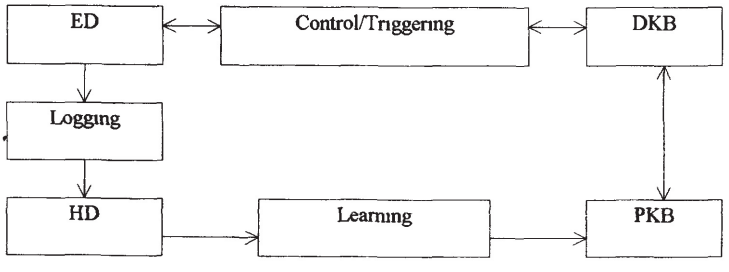

Fig. 5. Decision Process Model

Reasoning process

Each performance pattern is represented as a logical rule in PKB, and describes a correlation between the attributes of objects of stored grain disaster and related policies. These correlations come from HD, where selected attributes are logged according to the specific monitoring policy. If the status of the disaster objects satisfies the body part in the rule, the pattern from past experience represents that it is very likely that the status of the disaster object also satisfies

head port with some probability A logical rule has the generic form: IF X THEN Y, where X is its body part and Y is its head part. If some undesired status of a disaster object is foreseen to occur, it can further fire some logical rules in DKB and then trigger preventive control actions for stored grain disaster

Events

Managed object can be defined not only as a representation of a managed resource but also as part of the performance monitoring system The status of an object is a measure of its behavior at a discrete point in time and is represented by the current values of a set of status variables contained within a status vector. An event is an atomic entity that reflects a change in the status of an object. The status of an object has a duration in time; an event occurs instantaneously. Three kinds of events are of interest in stored grain system as follows.

(1) A control-flow event represents a control activity and is associated with a control thread.
(2) A data-flow event occurs when a status variable is changed or accessed.
(3) Process-level events show the creation and deletion of processes. They describe the interactions and data flow between processes.

Interfaces

The interface of a managed object can be divided into two parts as follows

(1) An operational interface supports the normal information processing operations, fulfilling the main purpose of the service provided by the object.
(2) A management interface supports performance monitoring and control interactions with the stored grain system.

Information generation

Monitoring information is generated by object instrumentation, where software or hardware probes or sensors detect events or generate status and event reports A sequence of such reports is used to generate a monitoring trace.

Status Reporting Status reports contain subsets of values from the status vectors and may include other related information such as time stamps and object identities Status reporting criteria define which reporting scheme to use, what the sampling period is, and the contents of each report.

Event Detection and Reporting. The detection of an event may be internal A function is used to update the status vector and to check the event-detection criteria An external agent can perform event detection. The agent receives status reports and detects changes in the state of the object. Once an event is detected, a report is generated that contains information such as the event identifier, type, priority, time of occurrence, and the state of the object before and after the event Event reports may also contain values of other application-specific status variables.

Trace Generation To describe the dynamic behavior of an object or a group of objects over a period of time, event and status reports are recorded in time order as monitoring traces. Such traces may be used for postmortem analysis. There are two kinds of traces. A complete trace contains all the monitoring reports generated by the system since the beginning of the monitoring session A segmented trace is a sequence of reports collected during a period of time

Information processing

After monitoring information is generated, it must be processed A monitoring service provides various functional units as follows. These units can be combined in different ways to suit the monitoring requirements

Trace Processing Monitoring traces may be constructed and ordered in various ways to provide different logical views of system activity over a period of time. The selection criteria in determining how monitoring traces are processed are in the followings.

(1) Generation, arrival time stamp, and other features of report.
(2) Identity, priority, and other features of reporting entity
(3) Identity and type of the managed object to which the report refers

Information Validation Performing validation and plausibility tests on monitoring information to make sure that the system has been monitored correctly is another important monitoring activity Validation is performed according to certain validation rules

Database Updating Valid monitoring information is used to maintain and update a representation of the current status of complex stored grain system. A conceptual database model of the stored grain system is constructed and continuously updated to represent the current status of the stored grain system There are two general approaches to collect the data for database updating/as follows

(1) Dynamic approach: user queries result in the automatic operation of relevant sensors in monitored objects, which collect the required data.
(2) Static approach: All possible monitoring data is to be collected and stored for potential access by users. The collection of data is independent of its use

Information Combination. The combination is to increase the level of abstraction of monitoring data. With the help of the process, users can observe the behavior of the stored grain system at a desired level of detail Thus, lower-level primitive events and states are processed and interpreted to

give a higher-level view of complex states and events

Filtering and analysis Modern monitoring systems of stored grain may produce thousands of alarms per day, making the task of real-time performance surveillance and fault management difficult. Due to the large volume of alarms, system operators frequently overlook or misinterpret them To reduce the number of alarms displayed on operators' terminal, current monitoring systems of stored grain apply alarm filtering procedures. On the other hand, monitoring systems of typical stored grain generate large amounts of monitoring information This results in heavy wage of computation resources. Filtering is the process of minimizing the amount of monitoring data so that users only receive desired data at a suitable level of detail

Dissemination and Presentation

Monitoring reports generated by objects are forwarded to different users of such information, including human users, managers, other monitoring objects and processing entities Several presentation techniques of Arc View 2.1 (ESRI, 1994) can be used in generalized systems to display configuration, performance and other information

Conclusions

One of effective methods for solving the problem of monitoring complex stored grain systems is to look into a new theory and related method for designing a framework for implementing the performance monitoring under complex stored grain environment Thus, an approach to monitor stored grain systems is developed to study the principle and application of monitoring model to design and implement the monitoring system. The monitoring model concept and its definition are two keys to design and implement the system. Research results for the model indicate that this new approach has also many other advantages such as simple model construction process, easy correctness verification, management and maintenance, and easy integration with other software packages such as expert system, machine learning systems, large database systems and special simulation systems A prototype system has been used to analyze and process a number of field data from certain stored grain system The final experiment resulting from the system shows that this model has very satisfactory performance and fast operation speed with comparison of traditional methods. Several improvements are possible For example, the approach to monitor complex stored grain systems can be used in conjunction with a Geographic Information System (GIS) in developing the initial monitoring model GIS is a promising tool for building complex monitoring system of stored grain. The major difficulty lies in creating a good architecture for the model implementation

Acknowledgments

The authors wish to acknowledge the valuable contributions from Mrs Chen Gui-Zhen to the research. They also acknowledge funding of both the Navy Research Project and the Project of China National Foundation of Natural Sciences (Project Number 39470415)

References

Aloni, O. et al. 1991 Performance Analysis of an Alarm Filtering Expert System World Cong. Exp Sys., Vol 4., pp. 2346 – 2354

Balzer, R et al 1982 Operation Specifications as the Basis for Rapid Prototyping. Software Eng Notes, Vol 10, No. 12, pp. 3 – 6.

ESRI. 1994 Introducing ArcView. Environmental Systems Research Institute Ltd , Redlands, CA.

Moffet, J. D and Sloman, M. S 1991 The representation of policies as system objects SIGOIS Bulletin, Vol 12, No 2 and 3, pp. 171 – 184.

Neumannm, P 1992 Risks to the public in computers and related systems. ACM Software Eng. Notes, Vol. 17, No. 1, pp 3 – 32.

Rellano, J et al 1991 GENESIS: An Expert System Shell for the Development of Symptom Pattern Recognition Expert Systems World Cong Exp Sys., Vol. 3 pp 1541 – 1549

Rumbaugh, J 1991 Object-Oriented Modeling and Design, Prentice-Hall, Englewood Cliffs, N J.

Snodgrass, R. 1988 A Relational Approach to Monitoring Complex Systems. ACM Trans on Computer Systems, Vol., 6, No 2, pp. 157 – 196

Wagenbauer, M. P. and Nejdl, W. 1993 Integrating Model-Based and Heuristic Features in a Real-Time Expert Systems. IEEE Expert, Intelligent Sys. and Their Applications, Vol 8, No 4, pp 1218 – 1228.