Designing Data Decision Framework on petroleum Field

Zhu Ping
Beijing Institute
KARAMAY HONGYOU Software CO., LTD.
Beijing, P. R. China
zhuping2404@sohu.com

***Abstract*—Data decision is one of the basic questions of sense computing and decision reasoning, there are not many works in the field of software design however. From the perspective of product development, this paper carries out a detailed study of data decision problems on petroleum field, according to business model, puts forward to generate the configuration file of decision processing, designs data decision framework based on the idea of model driven MDA and SOA architecture principles, and constructs integrated control platform for various scenes. The function design of the integrated control platform is also discussed. Finally, this paper summarizes the key points in the design of petroleum field data decision framework.**

Keywords—petroleum; data decision; framework; business model; MDA;

I. INTRODUCTION

By business logic modeling, the petroleum field business experts can model the basic process of petroleum field down hole engineering, i.e., drilling, recording, measuring and testing, and can describe in detail the large amount of uncertain data exception detection and data decision processes. Practical works need construct a general petroleum field data decision framework, automatically generate the main system source code and operation system and not realizing single decision system for specific problems, and need form a software platform products, according to the needs of business change, to build the different secondary development framework to save the workload for different scenarios of data decision product development. For example, the operating status exception detection and prediction system development of petroleum well pumping, the life prediction of the bit, and the equipment maintenance decision.

Data decision problems can be inducted as three kinds of problems: classification problem, decision analysis problem and optimization problem [1]. The data decision problems that this paper discusses are classification problem, which make the classification between normal and abnormal states. The methods which can solve the classification problems are: decision tree, support vector machine, wavelet analysis, clustering analysis and neural network. It is different to the above methods, the technology solutions that this paper deals with are experience rules which can be described by the expert, so by simple decision reasoning, the data decision problems can be solved. This way represents a large class of data decision problems. Our problem is how to use a unified development framework to simplify the development overload of data decision system, and how to form a unified data decision development platform.

At present, there are variety algorithms about the data decision [2,3,4,5], mostly in order to solve those complex problems that decision-making processes are not obvious. This paper deals with the decision problem, which decision-making processes are simple expert experience that can be a clear description by rules, therefore, using which kind of algorithms is not a core problem, the core problem is the system framework. Now, the architectures of the data decision support systems are data warehouse and on-line analysis processing mostly [6,7,8,9]. The problems that this paper deals with have the characters: stationary source data, decision problem can be described clearly by expert rule. Therefore, we suppose to use the expert modeling and model driven architecture (MDA) to drive the whole system framework, through the configuration properties, inheritance and component hot swap technology to achieve petroleum data decision framework development platform. In the next section of this article, we will discuss in detail the basic points of expert modeling; The third section, we will introduce the design idea of data decision framework platform; The fourth section, we will discuss examples of petroleum data decision system functions. Finally, we will summarize the full text and give further research work advice.

II. DATA DECISION COMPUTATION MODEL

We collect the experience of the petroleum data decision experts, and summarize the framework of the petroleum data decision problem. Each decision problem can be described as: decision problem ({features set}, {exception monitors set}, {equipment drivers set}, and {processing methods set}). The decision problem means single problem such as the gas detection decision, the exception prediction of the well leakage, exception prediction of the kick/blowout. For gas detection decision problem, the features set includes: time, well depth, total hydrocarbon content and moving average total hydrocarbon content, etc..The exception monitor reports state abnormal when the moving average of total hydrocarbon content is 1.5 times greater than the total hydrocarbon content, otherwise normal. The equipment driver means the sound and light alarm equipment driver. The processing method includes

the time, the processing position, the processing method character description and the disposal person.

Features are divided into three categories: system features, data localization features and decision attribute features. For example, for gas detection exception prediction decision problems, the time feature and the well depth feature in feature set are data localization features; the names of database and table where the field of total hydrocarbon content locates are system features; the total hydrocarbon content and the moving average total hydrocarbon content are the decision attribute features. Thus, the feature set can be described as:

feature set: ={feature |[feature set]};

feature:: =< system feature> <data locating feature> <decision attribute feature>;

system feature:: = <database><table>;

data location feature:: =< time ><well depth>;

decision attribute feature: =<total hydrocarbon content> <moving average total hydrocarbon content>;

Exception detector contains exception rule management module and interpreter module. Exception rule management module is responsible for the addition, deletion, modification, and query of the exception rules; the interpretation module is responsible for the identification, interpretation and execution of exception rules. Each decision problem has an exception detector, so each exception detector has ID for identifying the decision problem. Such as exception detecting rules:

IF (J.1.4.3.44.1.4> (J.1.4.3.44.1.3) *1.5) THEN 1 ELSE 0 ENDIF

Means: The average total hydrocarbon content of the mobile is 1.5 times higher than the total hydrocarbon content, otherwise normal. J.1.4.3.44.1.4 identifies the average total hydrocarbon content, and J.1.4.3.44.1.3 was identifies the total hydrocarbon content. Different decision rules are responsible for different levels of alert and alert reset. The exception detectors of all decision problems in the system constitute an exception detector set.

The set of the device drivers includes the driving interfaces of the devices, i.e. the sound and light alarm equipment, the alarm level Is divided into three levels alarm and is judged by different exception decision rules. For example, (J.1.4.3.44.1.4> (J.1.4.3.44.1.3)*1.5) and (J.1.4.3.44.1.4<= (J.1.4.3.44.1.3)*2.0) for the third level alarm; (J.1.4.3.44.1.4> (J.1.4.3.44.1.3)*2.0) and (J.1.4.3.44.1.4<= (J.1.4.3.44.1.3) *2.5) for the second level alarm; (J.1.4.3.44.1.4> (J.1.4.3.44.1.3)*2.5) for the first level alarm. The device drivers set also contains alarm reset interfaces and drivers, real-time database real-time interface API related to data decision problems, such as the real-time data access API interface and access method of the total hydrocarbon content.

The set of processing methods includes the starting time of the exception, the termination time of the exception, the processing suggestion, applying process method or not, the processing overview and the disposal person signature etc. The processing suggestion includes function modules as follows, rule management module and the interpreter module also, which is the same as the exception detector.

III. System Development Framework Design

First, we need to decide the principle of the system development framework:

(1) Design the system based SOA. SOA-based architecture design is the popular structure of WEB software, and has the features of simple operation and flexible configuration.

(2) Design the system based model driven MDA. All the functions are realized by the model as the configuration file, and the function is realized by MDA. This can enhance the scalability of the system.

(3) Design the system based on the open source framework. As far as possible the use of open source tool for development and implementation, it can greatly reduce the amount of engineering development, accelerate the development progress.

The applications of less access and concurrent users, can choose Tomcat, Jboss, Jfox and other open source projects as the operating environment, and it is very practical. The system with high graphical interface need, .NET can also be chosen as the operating container of the system. And the client application development prefers Eclipse, Netbeans, etc. as the running container of the system. For the basic framework, you can select PicoContainer, Spring and Avalon, etc. The performance layer can also choose Struts framework, which has been widely supported in many system development. In the model layer and data layer, the framework can be chosen, in addition to the famous entity EJB, as well as the lightweight persistence framework such as Hibernate, Toplink and Torque. for the system log management, you can chose log4j, XML parsing framework can chose Degester, OFBiz, workflow engine can chose shark, OBE or JBPM and content management can select cocoon which is open source framework that based on pipeline technology.

The development framework of the petroleum data decision system includes database, rule base, model base, data access service, log management, XML parsing service, rules management service, rules interpretation service, exception detection service, treatment suggestion service and decision processing service, etc. (as shown below).

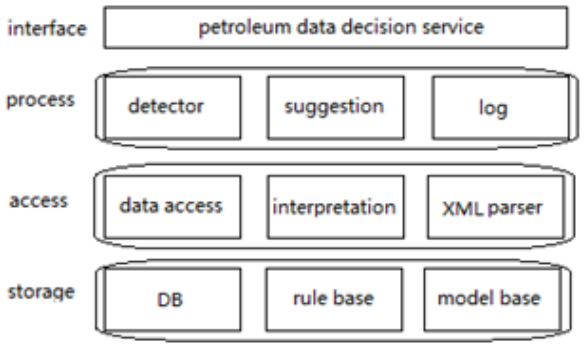

Fig.2 system development framework

The storage layer includes the real-time database, the exception rules base and the configuration models base and their management systems. The real-time database employs the mature products, such as PI at home and widely used in the power industry, IP21 in the petroleum field, PHD is more extensive used in chemical industry, PHD uses the Oracle relational database in internal; many small and medium-sized power enterprises chose eDNA. The exception rule base also takes the form of the file to store rules, which can be divided into the former part and the back part, where the former part elements can be an expression. The management of the exception rules base includes addition, deletion, modification, storage and query of rules and expressions. The standard exception rule is Horn clause: IF (logical expression) THEN 0 ELSE 1 ENDIF, of which 1 is abnormal, 0 is normal. The models base manages the engineering configuration file of various modules, service and system, includes the business model which is used for the system driven, and and the model of the exception judgment algorithm.

The access layer includes data access services, rule interpretation service, and the XML configuration file parsing service, etc. The information that the data access service needs XML file to configure includes database name, table name, API package, DDK package and SDK, API function name, parameter number and type, the data field names and its ID, etc. Its processes are realized by real-time data access services framework. Responsible for addition, deletion, modification, and query, the exception rules management and interpretation services store rules by the form of XML file, and for the interpretation of variables and expressions. The logic of the BNF description should be interpreted by the rules manager and interpreter as follows:

logical expression::=<logical variable>[<logical expression>][< constraint function >]

logical variable:: =<mathematical expression><relational operator><mathematical expression>.

relational operator::=<greater than>|<less than>|<equal to>|<greater than or equal to>|<less than or equal to>

mathematical expression::=<math unit> ><mathematical operator><math unit>

math unit::=< database field>|< mathematical constant>

constraint function: =<FUN(the exception duration)><relational operator><math expression>|<FUN(within a month)><relational operator><math expression>|<FUN(within a day)><relational operator><math expression>|<FUN (the times that the exception rules meet their conditions)><relational operator><math expression>

Among them, the mathematical variable is corresponding to exception feature value, and the mathematical constant is the numerical constants. The constraint function is usually implemented by the database storage process.

Our model base stores the XML configuration information of data access service, and the XML configuration information of rule manager and interpreter (said above BNF description), and the manager and interpreter read the BNF description to generate management and interpretation running frameworks. The model base also contains some special exception judgment algorithms, the model and system business logic, also contains the business identification information (This will be detailed in the follow-up article).

The process layer includes exception detection service, processing suggestion service and system log service. The exception detection service configures identification of business features, driven hardware identifications and their logical names, completes exception detection and hardware states setting function. The hardware driving function is completed configuration by the algorithm base driven with the model base. The processing suggestion service, according to the business logic configuration parameter, proposes exception process procedure, include process resume, processor, charger. Based on the open source framework, the system log service records the running statuses and events of the data decision framework systematically.

IV. SYSTEM FUNCTION DESIGN

The main functions of petroleum field data decision platform are: 1) the platform portal; 2) the function module of business model access; 3) data access module; 4) device driver module; 5) exception detection algorithm module; 6) rules analysis module; 7) real-time database interface module; 8) process suggestion module.

A. platform portal

First is the catalog of the petroleum field: drilling, recording, measuring, testing and so on, then leaf nodes are projects lists, for example: 1) the density adjustment decision of drilling fluid; 2) bit exception prediction decision; 3) well kick (blowout) exception prediction decision. Followed by the project status icon of decision project: normal/abnormal, then real-time data display, historical data query (including automatic query based on meta-data model and manual query by embedded SQL) and exception state records (exception occurrence time and exception end time, and duration); moreover is the configuration management (device driver, exception detection algorithm, the decision classification, project list and status icon). Finally, it is model adaptation management (to adapt between the multiple business models).

B. business model access module

Inputs business logic model, generates XML configuration files of services and modules. Including decision project identification and the required characters identification, called devices, drivers and related functions according to whether abnormal; requires automatic business logic input, integrity and correctness verification, and manual input function of business model. Business logic model is the foundation of service, system and module configuration file, after the business logic model produces configuration file, you can manually edit and save configuration file, and can preserve a record of the history of used configuration files, and can select default configuration file by the system menu.

C. data access module

Achieves real-time data by identifying the desired characters with the business model, this is achieved primarily through the real-time database API programming interface; mainly implements real-time database data transmission to the central database, requiring to realize the data access by the meta-data model automatically projected with the business model, manually input the meta-data model, and manually input SQL to access the central database by SQL. The central database can save the data of real time database (or index), and the data access of the central database is realized through the storage process. The XML configuration file of data access module includes real-time database path name, table name, field name, API functions, DDK library path, SDK library path, center database path name and business logic table etc.

D. device driver module

The device driver module includes the sound and light alarm devices, real-time database interfaces, exception process module and other device driver management programs. The system can dynamically hot swap configuration device, interface and processing module. The XML configuration file of the device driver module includes the identifications and interfaces of the sound and light alarm devices, the interfaces of the real-time database and the identification of the exception process module.

E. exception detection algorithm module

Realizes the exception detection algorithm base such as resolving regression equation algorithm; realizing the business logic rules exception constraints, such as less than a month, within a year, the number of N, which can be achieved by the stored procedure. The complete jar packet can be configured; the manual input rule, the constraint function base and its API are provided.

F. rule analysis module

Responsible for analyzing expert experience rules, recognizing the database field identification, identifying the API function name, mathematics and logical operator, and mathematical and logical expression, and the values of the mathematics and logic expression are calculated, finally, reaching conclusion of whether the alarm device is driven, and displaying the interface of the proposed processing suggestion. The rule analysis module strictly accords to the logical expression of the BNF paradigms, the rules which does not comply with the BNF paradigm will be feedback to the user in the rule verification phase.

G. real-time database interface module

Such as the PI real-time database, you can use interface program to control every operation who related PI database. These operations include building node and setting attribute configuration, data read and write and so on, so they can use PI-API to read and write data, and they can use PI-SDK to build node and set attribute configuration. You can also learn from OPC, the interface program is only responsible for data read and write, build node and set attribute configuration to the PI server by the database administrator manually. This way in the building PI node and setting attribute configuration, is much simpler: TAGNAME and INSTRUMENTTAG can be the same name, and data type, range, project units and deadband can be allocated according to the common sense and other attributes are available by default.

H. process suggestion module

Generate a process suggestion report, which uses a clear time granularity to generate one report.

V. SUMMARY

Decision problem resolving depends on data. In the actual production process, to obtain the priori data which decisions required, we often need to change decision flow, increased prenatal data investigation, trial production data collection, final product test data collection, and many other steps, form to the mathematical model of the decision problem, finally solve the mathematical problem. Data decision is the basis and premise of perception computation and decision - making. It is not only for petroleum field, but also for city field. In this paper, the solving method of data decision in the petroleum field is put forward in order to solve the problem of decision making of smart city.

Acknowledgment

Project supported by national 863 plan (2013AA01A607), the Xinjiang Uygur Autonomous Region major projects (201130110-3) and the Karamay Technology Bureau (SK2012-050).

References

[1] Wang Weihong, Qi Chao, Wei Yongchang, Li Bin, Zhu song, “On the review of the decision making method based on the data”, Journal of Automation, 2009,35 (6): 820-833

[2] Li Hong, Chen Songqiao, Zhao Rui, Guo Yuejian, “A multi-valued attribute and multi-class data decision tree algorithm”, Pattern Recognition and Artificial Intelligence, 2007,20 (6): 815-820

[3] Zou Baixian, “Research on the anomaly detection and prediction of network traffic”, the doctor degree paper of Graduate Institution of Chinese Academy of Sciences, August, 2003.

[4] Zhou Donghua, Wei Muhuan, Si Xiaosheng. “Research progress of industrial process exception detection, life prediction and maintenance decision”, Journal of Automation, 2013,39 (6) 711-722

[5] Jiang Shaohua, “Sealed blast furnace fault diagnosis and forecast research based on data-driven”, the doctor degree paper of Central South University, March 2009.

[6] Zhao Qiang, “Acceptance quality statistical analysis and system design based data decision”, the master degree paper of Tianjin University, May, 2011.

[7] Yu Wei, Wu Wenzhao, Li Huanjie, “Data decision-making system design for high education”, China Education Information, July 2008: 63-65

[8] Jiang Haitao, “The design and development of the financial data analysis and decision-making system”, the master degree of Jilin University, April, 2012.

[9] Pei Shenglei, “IDS prototype of data decision model and its application in mining industry”, Coal Technology, 2012,31 (9): 169-170