

Monitoring Data Publication System of Pollution Enterprise Based on Center Database of Province

Shulin Cheng^{1,*}, Wei Ye² and Meng Qian¹

¹School of Computer and Information, Anqing Normal University, Anqing Anhui, 246133, China

²Environment Information Center, Anqing Anhui, 246001, China

Abstract: As for the problem of publishing the monitoring data of pollution enterprises, one solution of monitoring data publication system is proposed based on the fusion of the provincial controlling database interfaces and the self-built database. The three-tier system architecture based on MVC framework is designed in B/S mode. Three key questions are analyzed and realized, including heterogeneous structure conversion of monitoring data, dynamic configuration of pollutant indicators, management and authentication of two-level rights. Lastly, a running instance is demonstrated to show that the system can effectively publish the monitoring data of pollution enterprises.

Keywords: Pollution Enterprises, Data Publication, MVC, Dynamic Indicators, Two-level Right Management

1. INTRODUCTION

Pollution source monitoring plays an important role in environmental controlling and protection, which is an important basis for environmental enforcement and environmental management. The city pollution source monitoring platform based on the center database of province is deployed in most medium-sized cities in China, which realizes pollution indicator data sampling, monitoring and statistics of the key pollution enterprises. The platform is mainly used by environmental management relevant departments. But the data has not been available to the public, and the general public does not know about the pollutant discharges of pollution enterprises. According to requirements of file of national Ministry of Environmental Protection in year of 2013 in China [1], it is necessary to further strengthen, promote and standardize the disclosure of monitoring information, which includes base information, monitoring results, monitoring positions, monitoring time, classes and concentration of pollutants, standard threshold, qualification, multiples over standards, discharge ways of pollutants and destination of discharge. Currently, monitoring data acquisition devices in key monitoring enterprises have been installed at the discharge ports, which collect and send the monitoring data of water pollutants and air pollutants to the control center of the province. The information will be stored in the database of province, and the monitoring information of other pollutants needs to be monitored by themselves. Therefore, the research of this paper is mainly on the design and implementation of monitoring data publication system of the key pollution enterprise based on the center database of province.

2. ANALYSIS OF MAIN PROBLEMS

The monitoring data of each city is stored in the center database of province, which includes two categories namely, monitoring data of pollution water and pollution air, which is collected and analyzed automatically by the data acquisition devices installed at the discharge ports, and sent to the center database of provinces through the network. The enterprise has no rights to obtain the data of automatic collection and alter them. Among of the data, hourly monitoring data and daily monitoring data need to be disclosed. For the pollution enterprises, the hour monitoring data is obtained from the center database of province, while daily data is obtained from the center database of province, if exists, maybe disclosed by themselves according to relevant technical specifications. The center database of province is not open considering the security and cannot be connected directly, but it provides the APIs of data acquisition. Therefore, the problem of different data storage and communication must be solved firstly in the publication system. Also, different enterprise stipulates different monitoring indicators according varying discharge pollutants which are showed in Table 1.

As can be seen from Table 1, the specific monitoring indicators and the number vary with different enterprises and different monitoring positions, the data publication system of monitoring data should be able to reflect the dynamic configuration of monitoring indicators. Also, the monitoring data in the center database of province is stored as vertical structure in accordance with the sampling time and pollution indicators, and APIs also provide vertical structure, while the monitoring data in actual publication system are required to be displayed in horizontal structure according to the sampling time and pollution indicators. The vertical structure and horizontal structure of monitoring data are showed in the following Table 2 and Table 3.

Table 1. Monitoring indicators sample of enterprises.

Type	Pollution Water Enterprise	
Enterprises	A Fine Chemical Co., Ltd.	A Paper Co., Ltd.
Monitoring Positions	Discharge Port 1	Discharge Port 1
Monitoring Indicators	PH, TP, COD, volatile phenol, total organic carbon, suspended solids, ammonia, total nitrogen compounds, sulfur compounds, petroleum	PH, suspended solids, biochemical oxygen demand, chemical oxygen demand, color, ammonia, total nitrogen, total phosphorus
Number	10	8
Type	Pollution Air Enterprise	
Enterprises	A Petroleum and Chemical Corporation	A Cement Co.
Monitoring Positions	Desulfurization exports of No.3, No.4 and No.5 furnaces	Incinerator chimney of Acrylonitrile Department
Monitoring Indicators	Soot, sulfur dioxide, nitrogen oxides	Sulfur dioxide, nitrogen oxide, acrylonitrile, cyanide
Number	4	3

Table 2. Vertical storage structure of hourly data of pollution water.

Data Fields Code	Data Type
PollutionCode	String
EmissionCode	String
PollutantCode	String
SampleTime	Datetime
AvgValue	Float
CorrectValue	Float
Operator	String
Department	String
CorrectTime	Datetime
Beiy	String

Obviously, there is heterogeneous problem between data structure in Table 2 and Table 3. And the indicators in Table 3 are dynamic, which are determined by the varying monitoring indicators in the discharge ports of enterprises. Therefore, the publication system should realize the transformation of heterogeneous structure in accordance with the configuration of dynamic indicators. Lastly, due to multiple user types in the publication system, each user may have different rights, and therefore the publication system should implement dynamic configuration of user rights and access authentication.

3. SYSTEM DESIGN

3.1. Architecture of Publication System

The publication system of monitoring data is built in the mode of B/S, which can be accessed by using browsers in client. The architecture of system is designed in the following Fig. (1).

The architecture of publication system comprises two parts namely, server side and client side. Server side includes database servers which subsume the center database of

Table 3. Horizontal display structure of hourly data of pollution water in publication system.

Sample Time	PH	COD	Suspended Solids	BOD	...
2014-02-02 10:00:00	5.80	29.38mg/l
...

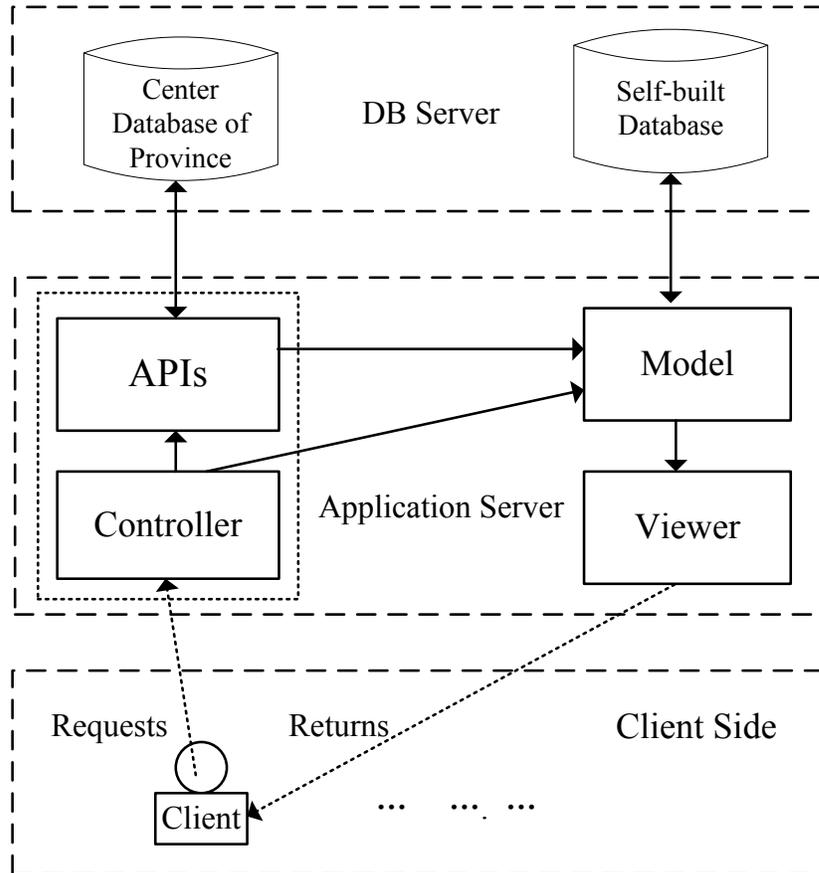


Fig. (1). Architecture of publication system.

province and self-built database, and application server. The center database of province stores all kinds of monitoring data transmitted from data acquisition devices of enterprises. The self-built database stores other monitoring information that needs to be disclosure, and mainly includes additional enterprises information, surrounding environmental monitoring information, monitoring information of noise, monitoring solutions, monitoring annual reports. Self-built database deployed in the cities, which maintains related information synchronization by calling the APIs provided by the center database of province. The synchronization information includes monitoring data, the name, code, discharge ports information, monitoring indicators of enterprise. The query APIs of provincial center database and the publication system are deployed in application server which is realized based on MVC framework [2-4] namely, Model-View-Controller. The input, business processing and output are separate but inner connected in MVC framework. The

publication system is designed and implemented by using Microsoft Visual Studio 2008 [5] and SQL Server 2005[6], and corresponding MVC framework is designed as LibDll, ASPNETViewer and CLibDll. MLibDll represents class library coded by C#, which are composed of three subclass libraries namely, data model subclass library, business rules subclass library and data access subclass library and they realized mapping of data storage tables, process business tasks and data access respectively. ASPNETViewer including ASP.NET Server controllers denotes the system interaction interfaces of input and output for displaying data. CLibDll coded by C# represents Controller class library for receiving inputs of users and calling corresponding programs of Model and View to accomplish users' requests. Client side is composed of HTML codes generated by the View programs at Server side. User can use browsers to access these HTML interfaces at client side.

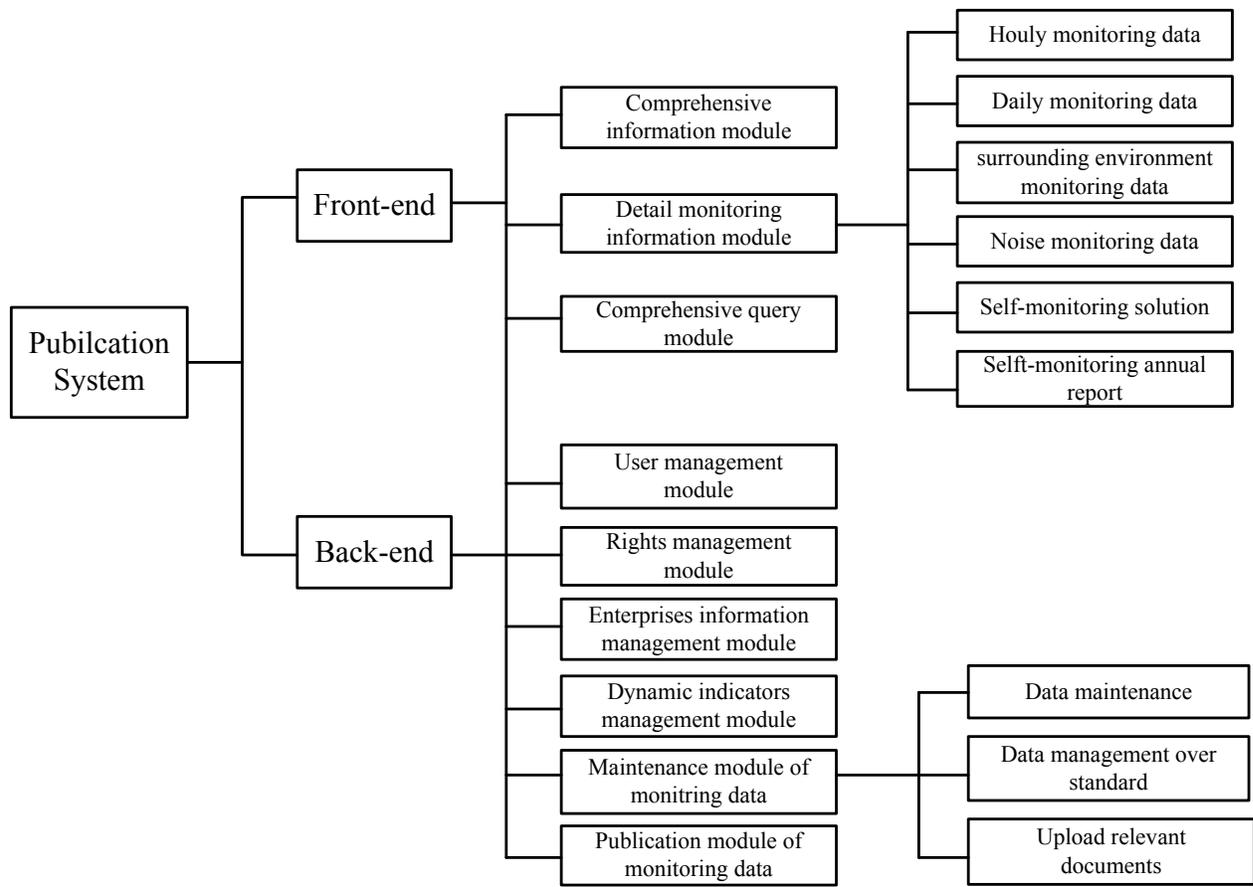


Fig. (2). Key modules of publication system.

3.2. Key Modules of Publication System

Monitoring data publication system is divided into two parts. One is front-end publication subsystem and the other is back-end data management subsystem. The key modules are as shown in Fig. (2).

Front-end publication subsystem mainly includes comprehensive information module, detailed monitoring information module and complex conditional query module. Comprehensive information module discloses the basis information of enterprises and the latest monitoring data. Detailed monitoring information module discloses the monitoring data of hourly data, daily data, environmental data, noise data, monitoring solutions and yearly reports. Complex conditional query module is responsible for querying the detailed monitoring data. Back-end subsystem mainly subsumes user management module, rights management module, enterprise information management module, dynamic index management module, monitoring data maintenance and distribution module. User management module realized the functions of registering new users, user information maintenance, password maintenance, setting user status. Rights management modules operated by system administrator grant rights to users and revoke rights from users. Enterprise information management module maintains the information of enterprises needed to be disclosed. Dynamic indicators management mainly sets the monitoring indicators needed to be public according to varying discharge ports of enterprises.

Monitoring data maintenance and publication modules mainly disclose and maintain the monitoring data responding to the indicators stipulated in advance. Among of the monitoring data, hourly data is automatically disclosed from the center database of province and not permitted to be altered, and daily data is disclosed automatically from some enterprises or disclosed manually for other enterprises in accordance with related administration files. Additionally, the rest monitoring information, such as surrounding quality data, is disclosed manually.

4. DESIGN AND IMPLEMENTATION OF SYSTEM KEY PROBLEMS

Enterprise monitoring data publishing system is an application of B/S mode, which is designed and implemented based on the center database of province. It should effectively solve not only the general questions but also the key problems discussed in Section 1. The solutions of general questions are relative simple and not discussed here.

4.1. Dynamic Indicators Controlling

A. Controlling Model Design

The monitoring indicators are different according to varying discharge ports of enterprises. Possibly, the indicators are also different even for the varying discharge ports of

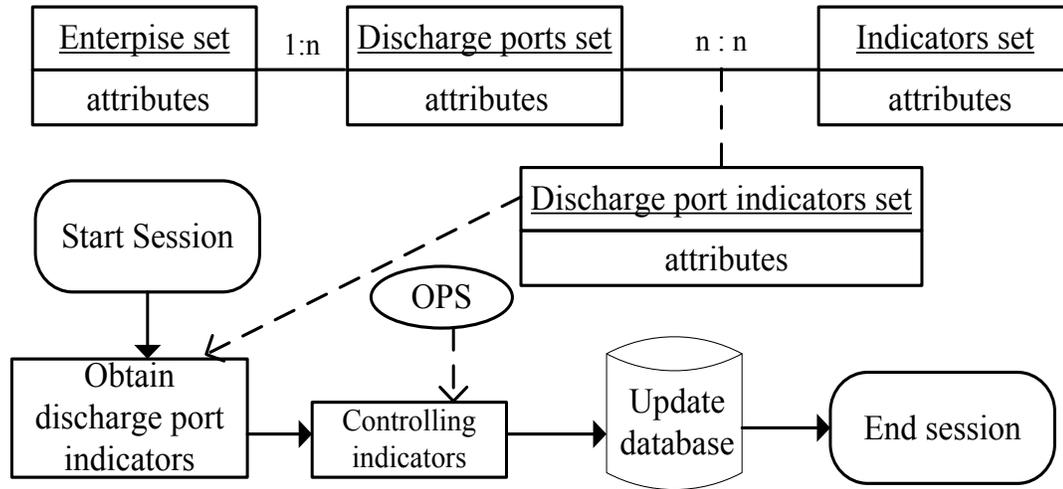


Fig. (3). Dynamic indicators controlling model.

the same enterprise. Therefore, the monitoring indicators configurations of each discharge port of each enterprise should be done beforehand in the publication system before disclosing monitoring data. The dynamic indicators controlling model is demonstrated in Fig. (3).

There is 1: n mapping relationship between enterprise entities and discharge port entities. That means one enterprise entity corresponds n discharge port entities and one discharge port corresponds one enterprise entity. Each discharge port has n different pollution indicators (namely monitoring indicators), and one pollution indicator belongs to n discharge ports, therefore discharge port set and pollution indicator set are n: n mapping relations. Each pollution indicator entity has many attributes including PollutantCode, DischargeCode, ShowSn. The discharge port indicators sets corresponds an indicator table in database. The table's keys are {PollutantCode, DischargeCode} which be utilized to dynamic pollutant indicators storage according to different discharge ports of varying enterprises. When starting a session and controlling the pollutant indicators of discharge ports, the program firstly obtains the set of indicators and then executes the corresponding operation such as get, insert, update and delete, according to the operation codes such as 00,01,02,03. The executed operation on indicator table makes its data be updated in database. The session ends.

B. Definitions of Basic Sets of Model

Let EntSet be enterprise set, $EntSet = \{p_1, p_2, p_3, \dots, p_m\}$, PDischargeSet be discharge ports set of enterprise, $PDischargeSet = \{(p_i, e_j) | p_i \in EntSet \wedge e_j \in DischargeSet\}$, DischargeSet be meta set of discharge ports set, $DischargeSet = \{e_1, e_2, e_3, \dots, e_n\}$, IndicatorSet be meta indicators set, $IndicatorSet = \{I_1, I_2, I_3, \dots, I_p\}$, DIndicatorSet be indicators set of discharge, $DIndicatorSet = DischargeSet \times IndicatorSet = \{(e_i, I_j) | e_i \in DischargeSet \wedge I_j \in IndicatorSet\}$. DIndicatorSet embodies n:n relationship between discharges and pollutant indicators namely, one discharge port has many pollutant

indicators and one indicator appears at the many discharge ports. As far as one enterprise is concerned, the discharge set of PDischargeSet should be set first, and then dynamic indicators set of DIndicatorSet can be set by selecting relevant indicators according to DischargeSet and actual pollutant indicators.

C. Main Relevant Model Functions

The function of Assign_Emissions(pi:string) is to realize the storage of discharge ports information of enterprise and return the discharge set of PDischargeSet.

The function of Get_Indicators(type:int) is to get pollutant indicators of pollution water or air for setting dynamic discharge indicators by calling APIs provided by the center database of province, and return IndicatorSet.

The function of Assign_Indicators(pi:string, ej:string, indicatorSet:list) implements the configuration and storage of dynamic indicators of discharge ports and return DIndicatorSet.

D. Program Controlling Strategies Between Monitoring Indicators and Monitoring Data

Since pollutant indicators of different discharge ports are varying, the program of system should generate the specific indicators according to the configuration of discharge indicators in database so that the program can capture the corresponding relationship between indicators and monitoring data and obtain the right monitoring data. So the adaptable controlling strategies between indicators and monitoring should be designed reasonably. The publication system is designed and implemented based on MVC framework. The View part is designed to dynamically generate input interfaces of pages in line with the discharge indicators got from database. In server side, the program obtains the IndicatorList object namely, DIndicatorSet defined in part C of this section, according to the parameters of enterprise code and

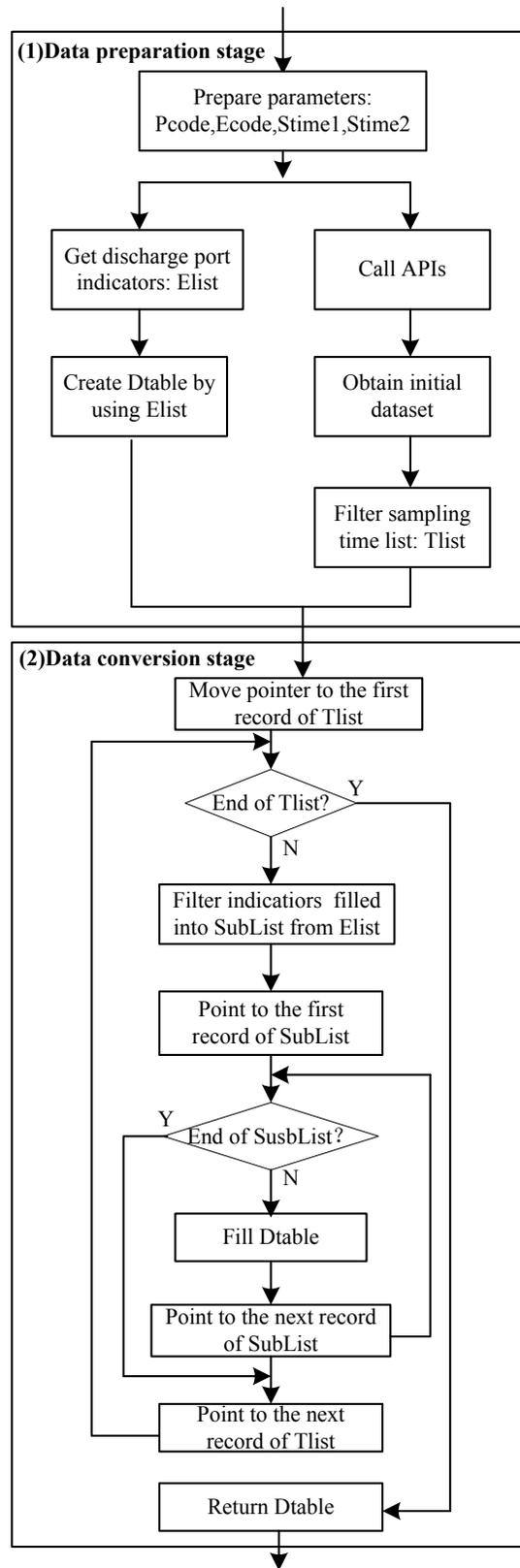


Fig. (4). Heterogeneous conversion flow between storage data and displaying data.

discharge code. Then the IndicatorList is utilized to create the Html<input> tag list which sent to the client browser. The one to one mapping relationship between Indicator List

and Input List guarantees the right submission of monitoring corresponding data the pollutant indicators.

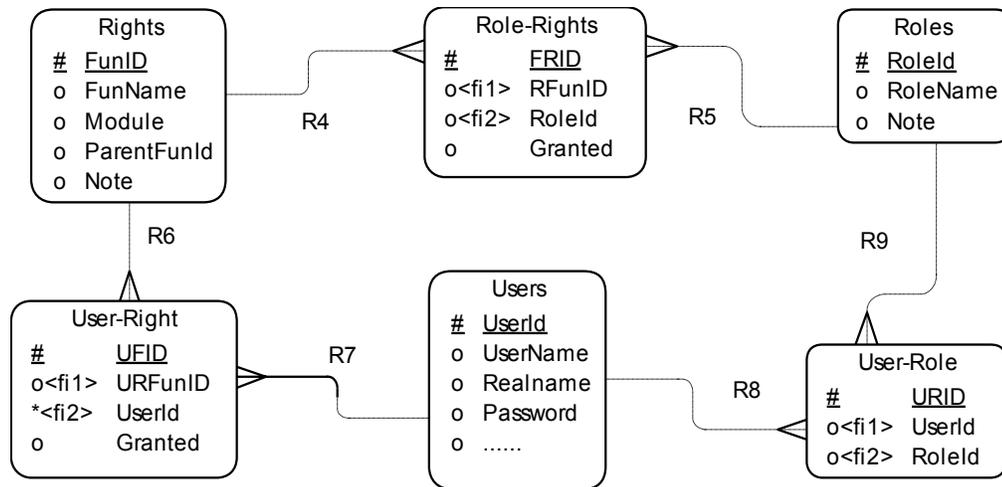


Fig. (5). Logic model of rights management.

4.2. Heterogeneous Conversion of Monitoring Data

From the discussions of section 2, we have known that heterogeneous problem of monitoring data exists between storage structure and displaying structure. Therefore, it is necessary to solve the heterogeneous conversion of monitoring data. The commonly used method is to first create temporary data table in database by using SQL Transaction commands, whose data structure is same as the display structure, and then to utilize the cursor commands and T-SQL sentences to fill the data into the temporary table [6]. But this method increases the burden of server. And mainly since most monitoring data is acquired from the center database of province and each enterprise cannot directly operate the center database of province except the data APIs, it is impossible for us to utilize the method. So we designed a two-stage conversion method namely, data preparation stage and conversion stage. First in preparation stage, the monitoring data needed to be transformed is obtained from the center database of province in line with the actual requirements, and the in the conversion stage the specialized application program is responsible for the heterogeneous conversion of monitoring data. The specific conversion flow is showed in Fig. (4). In data preparation stage, the initial monitoring data will be obtained from the center database of province by using APIs based on four parameters namely, Pcode, Ecode, Stime1 and Stime2, represent enterprise code, discharge code, and sampling time span respectively, and Tlist of sampling time list will be filtered from the initial data. Meanwhile, the Elist of pollutant indicators is obtained from the self-built database. Then the application program can create Dtable of monitoring data dynamically in memory according to Elist. Then the processing is getting into the second stage of heterogeneous data conversion. The relevant monitoring data corresponding to each sampling time record of Tlist and pollutant indicators in Elist will be filtered from initial data set. The data will be filled into Dtable one by one. The whole processing is carried out in an iterative way. After conversion, Dtable is the dataset we want and that can be easily displayed in web pages.

The algorithm of heterogeneous conversion is described in the following.

Input: Pcode, Ecode, Stime1, Stime2.

Output: DTable of dataset.

Step 1: Initiate looping variables of i, j , and Lists of Elist, Dtable, Tlist, SubList.

Step 2: Get Elist of discharge indicators, and the initiate dataset by calling APIs of center database of province, and filter the sampling time filled into Tlist.

Step 3: Move program pointer to the first record of Tlist.

Step 4: Judge whether the pointer arrives at the last record, if yes, go to step 7, otherwise build the subset SubList of discharge indicators according to Elist.

Step 5: Fill Dtable in accordance with Tlist and SubList

Step 6: Move program pointer to the next record of Tlist.

Step 7: return Dtable, end.

4.3. Two-Level Rights Authentication

There several roles in the publication system, such as role of administration user, role of common administration user, role enterprise user, role of data maintenance and role of public user. Each role of user has different rights which are set by specific system administrator. Very user should be authorized before using the publication system except browsing public data. Scalable rights management [7, 8] is implemented based on roles and two-level user rights authorization, which can grant rights and revoke rights easily, effectively and flexibly. The logic model is shown in Fig (5).

Each user maybe belongs to multiple roles and has all the rights of the roles. Of course, one role may correspond to multiple users. When system administrator grants right to a user, the role can be set first and then other rights of user are set. That means the rights of user are divided two levels. The rights of first level are from roles, and the second rights are from separate setting. If one user belongs to multiple roles, he/she will have all the rights of these roles. This kind of relationship between users, roles and rights can be seen in Fig. (5). There are three strengths of two-level rights management. One is flexible authorization. For instance, if 10

Latest Houly Data of Water			latest Daily Data of Water		
Publication Time	COD(mg/l)	AN(mg/l)	Publication Time	COD(mg/l)	AN(mg/l)
2014/2/18 20:00:46	29.8610	37.0280	2014/1/15 0:00:00	27.0000	19.0000
2014/2/18 19:00:46	30.2000	35.5660	2013/12/23 0:00:00	17.8000	30.3000
2014/2/18 18:00:46	25.2530	30.4940	2013/12/22 0:00:00	17.0000	23.0100
2014/2/18 17:00:46	25.0000	30.9010	2013/11/20 0:00:00	14.0230	29.1470
2014/2/18 16:00:46	32.6270	33.5460	2013/11/15 0:00:00	11.6910	34.9350
2014/2/18 15:00:46	32.7000	33.7930	2013/10/24 0:00:00	17.7870	33.6830

(a) Latest Monitoring Data Conversion

Smapling Time	AN(mg/l)	COD(mg/l)
2014/1/15 0:00:00	19.0000	27.0000
2013/12/23 0:00:00	30.3000	17.8000
2013/12/22 0:00:00	23.0100	17.0000
2013/11/20 0:00:00	29.1470	14.0230
2013/11/15 0:00:00	34.9350	11.6910
2013/10/24 0:00:00	33.6830	17.7870
2013/10/18 0:00:00	33.7500	18.4810

(b) Hour Monitoring Data Conversion of Multi-indicators

Fig. (6). The running sample of heterogeneous structures conversion.

users have the same rights, system administrator only set a new role with a name hypothesis of CommRole having these rights, then the 10 users can be set to CommRole. From the above analysis, we can know that the 10 users have the same rights. It's not necessary to set rights of user one by one. Certainly, if the 10 users have different rights, then the second level rights of users are set separately. The second is reuse of design-level and code-level, i.e., the logic model and corresponding codes could be reused in other designs and application programs. The third is better maintenance namely, this rights module is designed and realized as a stand-alone Webservice component which has outside uniform interfaces and is called by other programs and maintained separately. Therefore, the rights module in this publication system is deployed a group of Web Services named Au-WServices. Before user operates some business functions, the system sends the permission code and random security

code to server for checking. If passed, he/she will be authorized to operate the corresponding functions of system.

5. RUNNING PERFORMANCE OF PUBLICATION SYSTEM

The running sample of the publication system of monitoring data based on the center database of province is shown in Fig. (6). The figure shows conversion result of hourly monitoring data and daily monitoring data of pollution water discharge port in a Fine Chemical Corporation. Chart (a) of Fig. (6) shows the latest 6 records of hourly and daily monitoring data, and chart (b) of Fig. (6) gives the conversion result of querying hourly and daily monitoring data according to the sampling time span, discharge port and dynamic pollutant indicators. User can query in line with the combination of discharge port and indicators. Fig. (6) demonstrate

the good performance of heterogeneous monitoring data conversion and controlling of dynamic indicators in the publication system.

CONCLUSION

The monitoring data publication of pollution enterprise is the inevitable trend of future environmental management. Currently, the enterprises are required to disclose the monitoring data are mainly key pollution enterprises of state-controlling. In the future the enterprises of province-controlling and city-controlling are also required to disclose their monitoring data. And the data classes of monitoring data extend to real-time and ten-minute monitoring data. The publication system of monitoring is designed and implemented based on the center database of province and make full use of the combination of data APIs and self-built database to solve the key problems of system. The solution of the publication system of monitoring is significant and a reference for the further follow-up full disclosure of all monitoring data of all enterprises.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflicts of interest.

ACKNOWLEDGEMENTS

This work is supported by the Project of Department of Education of Anhui Province, China (No.KJ2011B087).

REFERENCES

- [1] "Files of Ministry of Environmental Protection". http://www.zhb.gov.cn/gkml/hbb/bwj/201308/t20130801_256772.htm.
- [2] H. Li, M. Zhou, G. J. Xu, and L. Si, "Aspect-oriented programming for MVC framework" *Biomedical Engineering and Computer Science (ICBECS), International Conference on IEEE*, pp. 1-4, 2010.
- [3] Z. Xie, H. He, Y. Li, and J. Jia, "Design and application of lighting energy consumption monitoring platform based on MVC and entity framework" *Information and Automation (ICIA), 2014 IEEE International Conference on IEEE*, pp.473-477, 2014.
- [4] M. R. Islam, M. M. Islam, and T. Halim, "A study of code cloning in server pages of web applications developed using classic asp. net and asp. net MVC framework" *Computer and Information Technology (ICCIT), 14th International Conference on IEEE*, pp.497-502, 2011.
- [5] K. Watson, J. V. Hammer, J. Reid, M. Skinner, D. Kemper, and C. Nagel, *Beginning Visual C# 2012 Programming*, John Wiley & Sons: USA, 2012.
- [6] P. Nielsen, and U. Parui, *Microsoft SQL Server 2008 Bible*. John Wiley & Sons: USA, 2011.
- [7] M. Strembeck, and J. Mendling, "Modeling process-related RBAC models with extended UML activity models," *Information and Software Technology*, vol.53, no.5, pp. 456-483, 2011.
- [8] M. U. Aftab, A. Nisar, D. Asif, M. Ashraf, and A. G. Burhan, *RBAC Architecture Design Issues in Institutions Collaborative Environment*". arXiv preprint arXiv:1310.5962, 2013.

Received: September 16, 2014

Revised: December 23, 2014

Accepted: December 31, 2014

© Cheng *et al.*; Licensee *Bentham Open*.

This is an open access article licensed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.