International Journal of Environmental Technology and Management

ISSN online: 1741-511X - ISSN print: 1466-2132
https://www.inderscience.com/ijetm

A risk detecting and preventing system for hazardous chemicals in energy industry based on knowledge graph

Guanlin Chen, Bangjie Zhu, Lei Zhou, Qi Lu, Wenyong Weng

DOI: 10.1504/IJETM.2023.10053932

Article History:

Received:	23 June 2022
Last revised:	08 August 2022
Accepted:	07 December 2022
Published online:	18 December 2023

A risk detecting and preventing system for hazardous chemicals in energy industry based on knowledge graph

Guanlin Chen, Bangjie Zhu and Lei Zhou

School of Computer and Computing Science,
Zhejiang University City College,
Hangzhou, 310015, China
Email: chenguanlin@zucc.edu.cn
Email: 31801305@stu.zucc.edu.cn
Email: 31701429@stu.zucc.edu.cn

Qi Lu

China National Air Separation Engineering Co., Ltd.,
Hangzhou, 310051, China
Email: luqi@cnasec.com

Wenyong Weng*

School of Computer and Computing Science,
Zhejiang University City College,
Hangzhou, 310015, China
Email: wengwy@zucc.edu.cn
*Corresponding author

Abstract: In order to improve the ability of hazardous chemicals management and risk prevention and control in the energy industry, combined with the concept of knowledge graph and visualisation related technology, this paper mainly introduces a risk detecting and preventing system for hazardous chemicals (RDPSHC). The system consists of user management, equipment management, visual display, log recording, real-time alarm, knowledge map application and so on. RDPSHC can extract entity relationships from a large number of hazardous chemical data and information, build knowledge graph, and provide corresponding treatment measures for hazardous chemicals according to the displayed visual early warning information, such as storage humidity and temperature of hazardous chemicals, so as to make the risk prevention and detection of hazardous chemicals more scientific and efficient and improve the supervision and prevention ability of hazardous chemicals in the energy industry.

Keywords: knowledge graph; hazardous chemicals; energy industry; Neo4j; visual display; Vue; Spring Boot.

Reference to this paper should be made as follows: Chen, G., Zhu, B., Zhou, L., Lu, Q. and Weng, W. (2024) 'A risk detecting and preventing system for hazardous chemicals in energy industry based on knowledge graph', *Int. J. Environmental Technology and Management*, Vol. 27, Nos. 1/2, pp.49–65.

Biographical notes: Guanlin Chen received his PhD degree from the Zhejiang University, China in 2013. Currently, he is a Professor in the School of Computer and Computing Science, Zhejiang University City College. His research interests include computer networks, big data and artificial intelligence.

Bangjie Zhu is a graduate student in the School of Computer and Computing Science, Zhejiang University City College. His research interests include artificial intelligence and smart city.

Lei Zhou is a graduate student in the School of Computer and Computing Science, Zhejiang University City College. His research interests include artificial intelligence and big data.

Qi Lu received his PhD degree from the Zhejiang University of Technology, China in 2010. Currently, he is a Senior Engineer in China National Air Separation Engineering Co., Ltd. His research interests include smart city and big data.

Wenyong Weng is an Associate Professor of Computer and Computing Science, Zhejiang University City College. His research interests include: big data processing and data mining.

1 Introduction

With the improvement and rapid development of the chemical industry chain in the national energy field, the types, quantity and scale of chemicals involved in the field of hazardous chemicals have also increased significantly. Limited by the complexity of hazardous chemicals storage conditions and the dispersion of regional storage, coupled with the difficulty of real-time linkage of urban energy data and information. It cannot meet the requirements of risk prevention and detection of hazardous chemicals in urban energy industry. Developing risk prevention and detection of hazardous chemicals based on knowledge graph technology and method is a new topic of hazardous chemicals in supervision in urban energy industry. Xu et al. (2020) proposed a Chinese sport knowledge graph based on Neo4j, the essence of knowledge graph is a knowledge graph with graph structure, it is a process of building graphics by extracting a large number of data entity relationships. Therefore, it is of great significance in the field of risk prevention and detection of hazardous chemicals in urban energy industry to develop a risk detection and prevention system of hazardous chemicals based on knowledge graph and construct a graphic database by extracting the entity relationship triples of hazardous chemicals data.

In recent years, some research on the application of risk prevention and detection management and knowledge graph of hazardous chemicals has gradually emerged. Xiaogang and Xin (2011) proposed the research on hazardous chemicals information management system. Kertkeidkachorn and Ichise (2017) proposed an end-to-end system to create a knowledge graph from unstructured text. He et al. (2020) proposed the knowledge graph augmented word representation of named entity recognition. In the same year, Yunlong et al. (2020) provided a scheme for early warning of special work risks based on event knowledge graph. Zhang et al. (2020) also proposed the construction method of marine hazardous chemicals knowledge graph based on IMDG rules. Zhou et al. (2021) proposed an intelligent management system to deal with the management of hazardous chemicals park, which studied the management of hazardous chemicals park.

The main content of this paper is to design and implement a risk detecting and preventing system for hazardous chemicals (RDPSHC) in energy industry based on knowledge graph. By extracting the entity relationship from a large number of hazardous chemical data and constructing the corresponding knowledge graph, the risk decision-makers of hazardous chemicals in the energy industry can better manage and prevent the potential risks of hazardous chemicals in the form of visual charts, improve the improvement efficiency of relevant analysis, and provide more scientific and efficient decision-making help for the risk prevention and detection of hazardous chemicals in the urban energy field.

In 2020, Wang et al. proposed a health status reporting system based on Spring Boot, *RDPSHC* is developed based on this basic framework, and adopt the mainstream front and back-end separation Vue and Spring Boot development framework. With the help of knowledge graph technology, the system uses graph database Neo4j and MySQL database, element UI and Vuetify plug-in to simplify the code complexity, and based on Shen and Wang (2021) provided a visualisation tools and forms to present system, *RDPSHC* uses the ECharts visualisation tool to build a triplet of structured dangerous goods information and load it into the database in the form of graph. The system will give a real-time alert according to the monitoring data; provide good decision support for decision-makers of risk prevention and detection of dangerous chemicals.

2 Overall system design and analysis

2.1 Overall system design

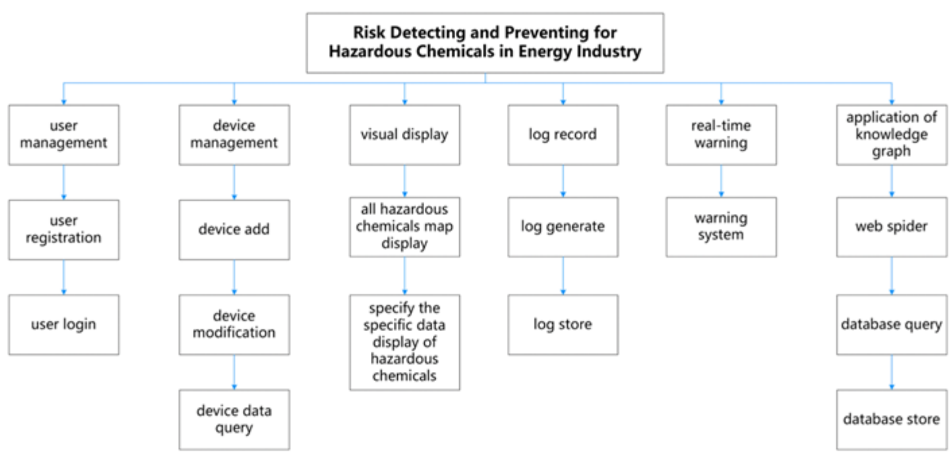
The main task objective of the system is to timely warn the existing or potential environmental risks in the storage and management of energy related hazardous chemicals, extract the entity relationship of simulated hazardous chemicals data, establish corresponding graphical database, set environmental risk threshold, provide selection suggestions and countermeasures to relevant decision-makers or units, and improve the efficiency of hazardous chemicals management and risk detection in the energy industry.

RDPSHC is composed of six modules, and the basic function framework is shown in Figure 1:

- User management: it includes the functions of user addition, user password modification and user login. The system does not support the direct registration of users on the login panel. The registration needs to be operated by logging in to the existing administrator account.
- Device management: it supports the addition and modification of the monitoring device. The system also supports the screening and query of historical data of the specified monitoring device, monitoring type and monitoring time period. The data information is displayed in the form of charts.

- Visual display: display the information of all monitoring points on the map in the form of icons, support the designated monitoring device, and display the real-time data and historical data statistics of the corresponding monitoring device.
- Log record: log the monitoring data and store it in the database.
- Real-time warning: the hazardous chemicals information database constructed by using the knowledge graph technology and the existing open source knowledge graph information set on the network, combined with the real-time data of the monitoring device, warn the potential storage risk of hazardous chemicals.
- Application of knowledge graph: the information and materials stored in hazardous chemicals are captured by web crawlers. At the same time, combined with the structured knowledge graph information API provided by ownthink, the largest open-source Chinese knowledge graph database, a triplet is established, and finally the data is exported to the database for real-time early warning module.

Figure 1 The basic framework of RDPSHC (see online version for colours)



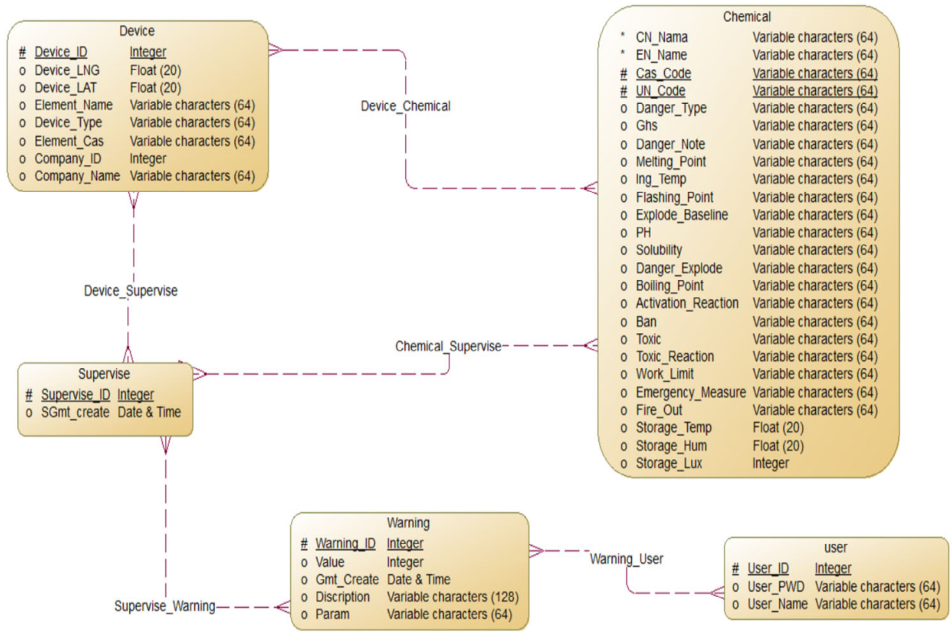
2.2 Database design

In 2019, Zhao et al. proposed an analysis of disease data based on Neo4j graph database, on this basis; RDPSHC adopts MySQL relational database and Neo4j graph database, and designs five data tables, namely user table, device information table, chemical information table, supervision information table and warning information table.

The database structure of this system is shown in Figure 2.

The relationship between the above data tables is described as follows: the supervise information table refers to the device and chemicals information table, provides the monitoring results, is referenced by the warning information table, and is finally referenced by the user table to provide the user with alert information.

Figure 2 The design of the system database (see online version for colours)



3 Detailed system design and implementation

RDPSHC uses the development framework of front and back-end separation Vue and Spring Boot, combined with MVC design mode, and uses graph database Neo4j and relational database MySQL to store corresponding data and triplet information, so as to manage the potential risk of hazardous chemicals and give real-time warning. For example, the system integrates the knowledge graph structured data model method to realise the semi-automatic way to convert the data into triples and store them in the graph database. With the help of the information base and external API interface provided by ownthink, the largest open-source Chinese knowledge graph database at present, the characteristics of corresponding hazardous chemicals and corresponding treatment measures are displayed; display of visual monitoring data through ECharts plug-in; use map positioning, log generation and other operations to provide alert prompt functions.

The following mainly describes in detail the specific implementation of key modules such as the establishment of knowledge graph and structured data, visual data display and alert prompt mechanism in the system.

3.1 Design and implementation of knowledge graph application

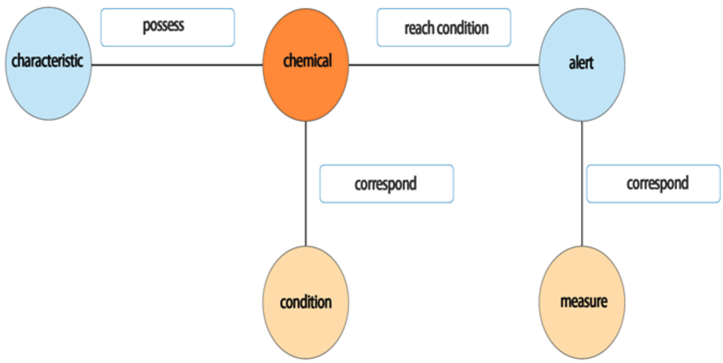
The application of knowledge graph adopts the method of capturing structured data to build a small knowledge graph locally. Firstly, the self built crawler program is used to obtain the relevant attributes and text descriptions of different hazardous chemicals, and then the data is semi automatically converted into triples and stored in the graph database.

In order to improve the data richness of the knowledge graph, the system also adopts the open source Chinese knowledge graph database ownthink information database.

The external API interface is used to display the alert information of the system and the sorted and summarised characteristic information of structured hazardous chemicals, including the storage mode and emergency treatment measures of hazardous chemicals, so as to assist the user in handling and troubleshooting the alert.

The important model elements of the knowledge graph established by the system are shown in Figure 3.

Figure 3 The knowledge graph model diagram (see online version for colours)



Based on the knowledge graph model, hazardous chemicals, early warning indicators (such as storage ambient temperature, humidity, etc.) and measures are constructed into a triple. Hazardous chemicals and measures are regarded as two entities. Early warning indicators mean that when hazardous chemicals reach the threshold of the indicator, corresponding measures need to be taken to solve and complete the process of early warning mechanism. In the extraction of entity attributes, hazardous chemicals are regarded as entities and have an early warning indicator attribute, that is, when the early warning indicator attribute reaches the threshold, its attribute value is the corresponding measure.

In this mode, through a large number of triples to quickly build a knowledge graph, using the graph database entity association and a large number of data compatibility, as well as the advantages of supporting rapid query of associated data, we can realise the equipment sensing early warning indicators, quickly match hazardous chemicals and give corresponding solutions, and realise the early warning mechanism.

Important codes involved in building the knowledge graph are as following:

```
#Crawler code logic, load the necessary library first, and omit it here
fileName = 'News.csv'
wd = webdriver.Chrome()
wd.get("http://www.ccin.com.cn/c/index_domestic") # Open the page you want to grabfor i in range (1, 5):
    time.sleep(2) # Set the waiting time until the web page is loaded # Using XPath to locate structured data information
```

```
wd.find_element_by_xpath("/html/body/div[3]/div/div[1]/div/div[2]/div[6]").click( )
soup = BeautifulSoup(wd.page_source, "lxml")
# Scan all secondary links of the list page and put them into the array for circular traversal
for link in soup.find_all(name='a', attrs={"href":re.compile(r'^/detail')}):
    if flag == 0:
        list.append('http://www.ccin.com.cn' + link.get('href'))
        flag = 1
    else:
        flag = 0
for element in list:
    mydict = {}
    wd.get(element)
    elementsoup = BeautifulSoup(wd.page_source, "lxml")
    title = elementsoup.find(name='h1')
    content = elementsoup.find('div', class_= 'news-content-txt')
    # Grab the data content on the web page and store it in the dictionary
    mydict["title"] = title.text
    mydict["content"] = content.get_text
    resultlist.append(mydict)
data = pd.DataFrame(resultlist)
# Then save the data into CSV file
try:
    if number == 1:
        csv_headers = [' title ', ' content ']
        data.to_csv(fileName, header=csv_headers, index=False, mode='a+', encoding = 'utf_8_sig')
    else:
        data.to_csv(fileName, header=False, index=False, mode='a+', encoding= 'utf_8_sig')
        number = number + 1
    except UnicodeEncodeError:
```

```
    print("Coding error. The data cannot be written to the file. Ignore the data directly")
    time.sleep(10)
    wd.quit() # Close browser
// JiebaWord segmentation
JiebaSegmenter segmenter = new JiebaSegmenter();
    String[] sentences =
            new String[] {text};
        int topN=5;
        TFIDFAnalyzer tfidfAnalyzer = new TFIDFAnalyzer();
        for (String sentence: sentences) {
            System.out.println(segmenter.process(sentence,
            JiebaSegmenter.SegMode.SEARCH).toString());
            List<Keyword> list=tfidfAnalyzer.analyze(sentence,topN);
            for(Keyword word:list)
                System.out.println(word.getName()+":"+word.getTfidfvalue()+",");
    }
```

Through the above construction of knowledge graph logic code, the data records of structured hazardous chemicals are shown in Figure 4.

The API query interface provided by the Chinese knowledge graph database ownthink is quoted to add the information of hazardous chemicals into the alert packaging information. The code is as following:

```
@Async // Turn on Multithreading
    @Scheduled(cron = "0/5 * * * * ?") // Set scheduled tasks
public void generateData() throws InterruptedException{
    List<DeviceVO> deviceVOList = deviceService.getDeviceList();
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
    log.info(sdf.format(new Date()) + ", processed by thread:
    "+Thread.currentThread().getName()+" reading data");
    for(DeviceVO deviceVO: deviceVOList){
        DeviceDataVO deviceDataVO = generateCustomize(deviceVO);
        if(deviceDataVO != null){
            Integer deviceId = deviceVO.getDeviceId();
            if((deviceDataVO.getTemp()>=deviceVO.getStorageTemp()) &&
            deviceVO.getStorageTemp()!=-999){ String description = " Warning!
```

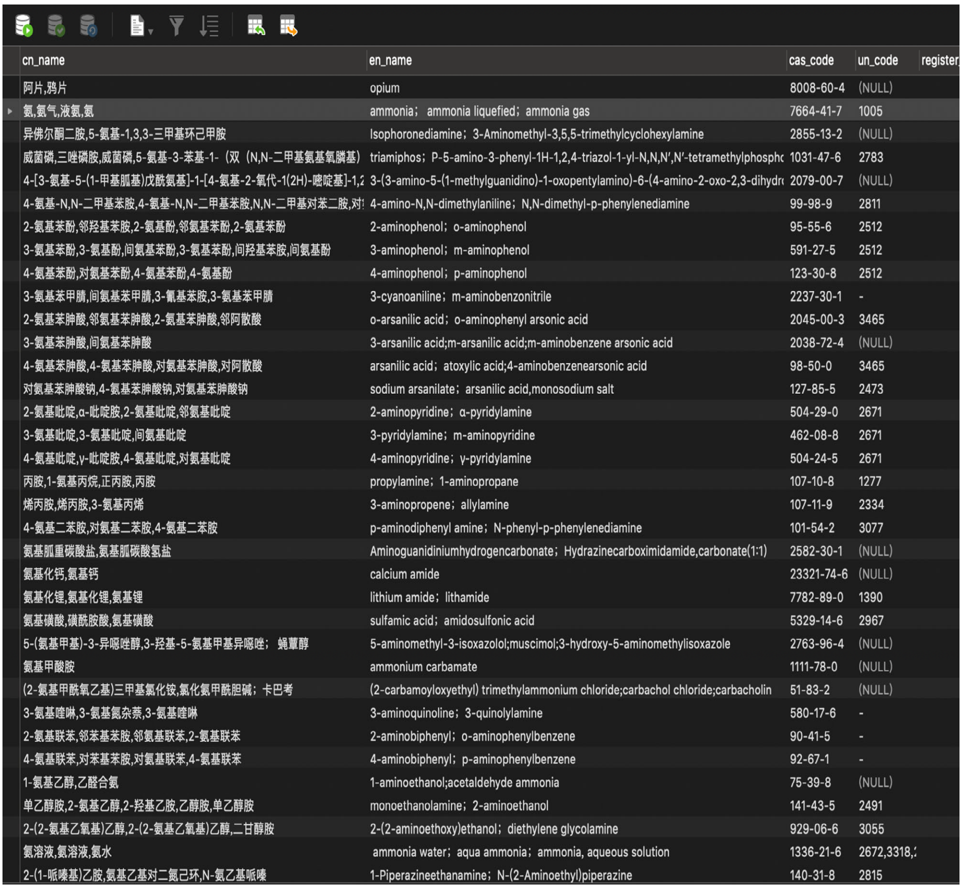
```
        Hazardous chemicals monitored by Device "+deviceId+":
        "+deviceVO.getElementName()+",storage humidity exceeds threshold,
        current temperature is "+deviceDataVO.getTemp()+"°C, please check!";
            log.info(description);
// Encapsulate the hazardous chemicals information provided by API WarningInfoDO
warningInfoDO = extractDataIntoWarningDO(deviceVO, deviceDataVO,
"temperature", deviceDataVO.getTemp());
            warningInfoDO.setDiscription(description);
            warningInfoDO.setInfo1(infoJsonObject.get("store");
            warningInfoDO.setInfo2(infoJsonObject.get("CAS ID");
            dataService.insertWarningRecord(warningInfoDO);
            continue;
        }
        boolean todayDataExisted = dataService.todayDataExisted(deviceId);
        deviceDataVO.setDeviceId(deviceId);
        deviceDataVO.setGmtCreate(new Date());
        if(!todayDataExisted){
            dataService.saveRecord(deviceDataVO);
        }
    }
  }
}
```

3.2 Design and implementation of visual display module

The monitoring data of the device of the hazardous chemicals risk prevention and detection system is displayed to the user in the form of graphic visualisation. The monitoring data fed back from the back end is displayed intuitively by using the ECharts visualisation tool plug-in in the form of charts, tables and text. The option attribute is configured through the visual chart provided by ECharts. Taking the design of the temperature chart of the monitoring device as an example, the ECharts configuration is called temperatureOption. Temperature monitoring needs to obtain the value of the page variable temp, which is assigned to this after the data acquisition method of monitoring device executed by the scheduled task, assign the temperature information to this temperatureOption. series[0].data. Value, ECharts will read the variable information. For the gradient ring of temperature data, add the series array element in the temperatureOption. The new this.$echarts.LinearGradient method sets the gradient element. The specific gradient colour is set through the colour of axisline and linestyle. The gradient area and shape can be modified. The corresponding data text information display can be set in the formatter of the detail field under the series attribute. The configuration logic of the dot pointer is the same as above. Dot shape can be controlled by setting fields such as startAngle, clockWise and radius.

The visual display of the system is shown in Figure 5.

Figure 4 The structured data diagram (see online version for colours)



cn_name	en_name	cas_code	un_code	register
阿片,鸦片	opium	8008-60-4	(NULL)	
氨,氨气,液氨,氨	ammonia； ammonia liquefied；ammonia gas	7664-41-7	1005	
异佛尔酮二胺,5-氨基-1,3,3-三甲基环己甲胺	Isophoronediamine；3-Aminomethyl-3,5,5-trimethylcyclohexylamine	2855-13-2	(NULL)	
威菌磷,三唑磷胺,威菌磷,5-氨基-3-苯基-1-（双（N,N-二甲基氨基氧膦基）	triamiphos；P-5-amino-3-phenyl-1H-1,2,4-triazol-1-yl-N,N,N′,N′-tetramethylphosph	1031-47-6	2783	
4-[3-氨基-5-(1-甲基胍基)戊酰氨基]-1-[4-氨基-2-氧代-1(2H)-嘧啶基]-1,2	3-(3-amino-5-(1-methylguanidino)-1-oxopentylamino)-6-(4-amino-2-oxo-2,3-dihydr	2079-00-7	(NULL)	
4-氨基-N,N-二甲基苯胺,4-氨基-N,N-二甲基苯胺,N,N-二甲基对苯二胺,对	4-amino-N,N-dimethylaniline；N,N-dimethyl-p-phenylenediamine	99-98-9	2811	
2-氨基苯酚,邻羟基苯胺,2-氨基酚,邻氨基苯酚,2-氨基苯酚	2-aminophenol；o-aminophenol	95-55-6	2512	
3-氨基苯酚,3-氨基酚,间氨基苯酚,3-氨基苯酚,间羟基苯胺,间氨基酚	3-aminophenol；m-aminophenol	591-27-5	2512	
4-氨基苯酚,对氨基苯酚,4-氨基苯酚,4-氨基酚	4-aminophenol；p-aminophenol	123-30-8	2512	
3-氨基苯甲腈,间氨基苯甲腈,3-氰基苯胺,3-氨基苯甲腈	3-cyanoaniline；m-aminobenzonitrile	2237-30-1	-	
2-氨基苯胂酸,邻氨基苯胂酸,2-氨基苯胂酸,邻阿散酸	o-arsanilic acid；o-aminophenyl arsonic acid	2045-00-3	3465	
3-氨基苯胂酸,间氨基苯胂酸	3-arsanilic acid;m-arsanilic acid;m-aminobenzene arsonic acid	2038-72-4	(NULL)	
4-氨基苯胂酸,4-氨基苯胂酸,对氨基苯胂酸,对阿散酸	arsanilic acid；atoxylic acid;4-aminobenzenearsonic acid	98-50-0	3465	
对氨基苯胂酸钠,4-氨基苯胂酸钠,对氨基苯胂酸钠	sodium arsanilate；arsanilic acid,monosodium salt	127-85-5	2473	
2-氨基吡啶,α-吡啶胺,2-氨基吡啶,邻氨基吡啶	2-aminopyridine；α-pyridylamine	504-29-0	2671	
3-氨基吡啶,3-氨基吡啶,间氨基吡啶	3-pyridylamine；m-aminopyridine	462-08-8	2671	
4-氨基吡啶,γ-吡啶胺,4-氨基吡啶,对氨基吡啶	4-aminopyridine；γ-pyridylamine	504-24-5	2671	
丙胺,1-氨基丙烷,正丙胺,丙胺	propylamine；1-aminopropane	107-10-8	1277	
烯丙胺,烯丙胺,3-氨基丙烯	3-aminopropene；allylamine	107-11-9	2334	
4-氨基二苯胺,对氨基二苯胺,4-氨基二苯胺	p-aminodiphenyl amine；N-phenyl-p-phenylenediamine	101-54-2	3077	
氨基胍重碳酸盐,氨基胍碳酸氢盐	Aminoguanidiniumhydrogencarbonate；Hydrazinecarboximidamide,carbonate(1:1)	2582-30-1	(NULL)	
氨基化钙,氨基钙	calcium amide	23321-74-6	(NULL)	
氨基化锂,氨基化锂,氨基锂	lithium amide；lithamide	7782-89-0	1390	
氨基磺酸,磺酰胺酸,氨基磺酸	sulfamic acid；amidosulfonic acid	5329-14-6	2967	
5-(氨基甲基)-3-异噁唑醇,3-羟基-5-氨基甲基异噁唑；蝇蕈醇	5-aminomethyl-3-isoxazolol;muscimol;3-hydroxy-5-aminomethylisoxazole	2763-96-4	(NULL)	
氨基甲酸胺	ammonium carbamate	1111-78-0	(NULL)	
(2-氨基甲酰氧乙基)三甲基氯化铵,氯化氨甲酰胆碱；卡巴考	(2-carbamoyloxyethyl) trimethylammonium chloride;carbachol chloride;carbacholin	51-83-2	(NULL)	
3-氨基喹啉,3-氨基氮杂萘,3-氨基喹啉	3-aminoquinoline；3-quinolylamine	580-17-6	-	
2-氨基联苯,邻苯基苯胺,邻氨基联苯,2-氨基联苯	2-aminobiphenyl；o-aminophenylbenzene	90-41-5	-	
4-氨基联苯,对苯基苯胺,对氨基联苯,4-氨基联苯	4-aminobiphenyl；p-aminophenylbenzene	92-67-1	-	
1-氨基乙醇,乙醛合氨	1-aminoethanol;acetaldehyde ammonia	75-39-8	(NULL)	
单乙醇胺,2-氨基乙醇,2-羟基乙胺,乙醇胺,单乙醇胺	monoethanolamine；2-aminoethanol	141-43-5	2491	
2-(2-氨基乙氧基)乙醇,2-(2-氨基乙氧基)乙醇,二甘醇胺	2-(2-aminoethoxy)ethanol；diethylene glycolamine	929-06-6	3055	
氨溶液,氨溶液,氨水	ammonia water；aqua ammonia；ammonia, aqueous solution	1336-21-6	2672,3318,	
2-(1-哌嗪基)乙胺,氨基乙基对二氮己环,N-氨乙基哌嗪	1-Piperazineethanamine；N-(2-Aminoethyl)piperazine	140-31-8	2815	

The realisation of the system switching monitoring device function: the front end reads the device list and selects the device ID. The front end applies for reading the real-time information of the binding ID every five seconds through the timer and displays it through the visualisation module. The switching of device monitoring data is shown in Figure 6.

3.3 Design and implementation of alert mechanism

The main realisation of the alert prompt mechanism is that when the data obtained by the monitoring device exceeds the environmental threshold of the corresponding hazardous chemicals storage, the back-end will feed back to the alert monitoring interface, generate log information and store it in the database. The front end carries out prompt box warning according to the feedback information, in which the characteristic information and corresponding treatment measures of corresponding hazardous chemicals are displayed in the prompt box by using the technology of knowledge graph. Alert content mainly includes map positioning, names of hazardous chemicals, device monitoring data and relevant characteristic information.

Figure 5 The visual display diagram (see online version for colours)

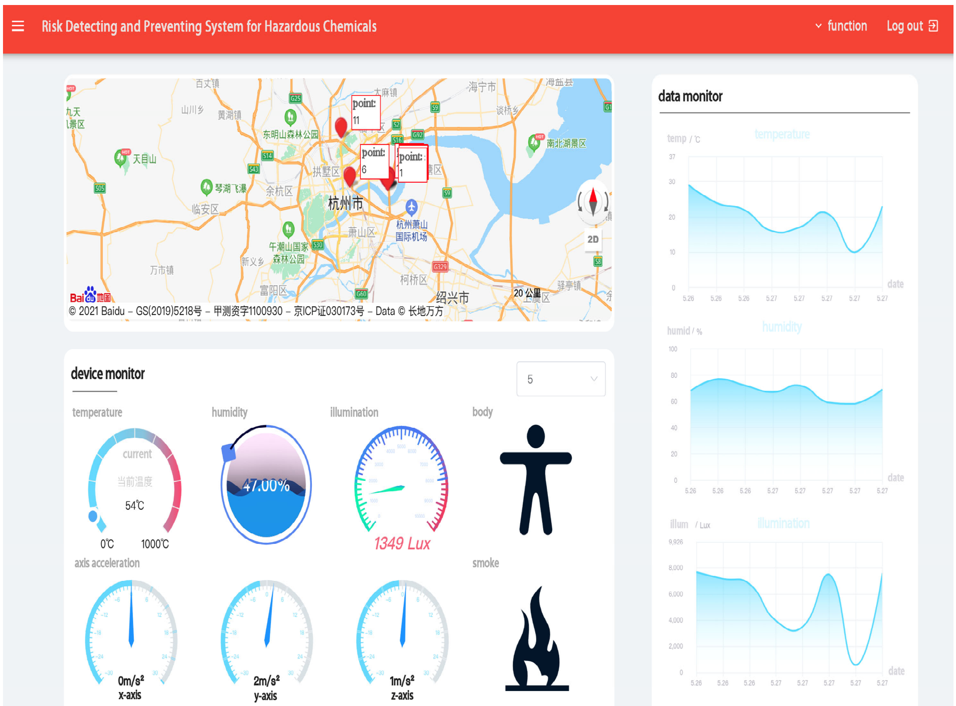
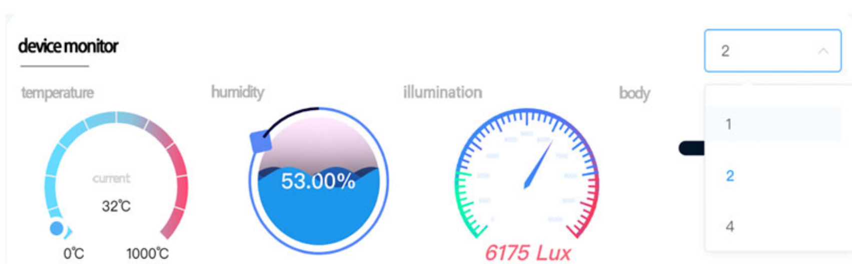


Figure 6 The device data switching diagram (see online version for colours)



The prompt alert box combines map positioning and knowledge graph content processing to facilitate decision-makers to know the geographical location, characteristic information and emergency measures of hazardous chemicals on alert for the first time. The alert prompt interface is shown in Figure 7.

Figure 7 The warning diagram (see online version for colours)

Figure 8 The alert record diagram (see online version for colours)

Alert record information is summarised in the alert record interface to display the number of alerts and the list of alert records in recent one week. Alert records are displayed in reverse chronological order according to the alert list information returned by the back end. The user can use the time selector to filter the records. When a specified record is selected, the front end reads the corresponding alert details, updates the map geographical location, and displays the specific alert location. At the same time, with the help of EasyExcel tool, the alert records can be exported in tabular form.

The alert recording interface is shown in Figure 8.

The logic code for switching map geographic information of selected alert record is as follows:

```
mounted() {
    this.mapinit()
    },
    watch: {
        // Listen to the alarm list, select changes and reinitialise the map activeWarning:
        {
        handler() {
            this.mapinit()
            },
            immediate: true,
            deep: true
            }
        }
mapinit() {
        var map = new BMap.Map("pos-map");
        // Take the location of the currently selected alarm as the centre
        var point = new BMap.Point(this.activeWarning.deviceLng,
        this.activeWarning.deviceLat);
        // Set map centre and zoom
        map.centerAndZoom(point, 17);
        var marker = new BMap.Marker(point); // Create dimensions
        map.addOverlay(marker);// Add dimensions to your map
        map.enableScrollWheelZoom(true);
        var myGeo = new BMap.Geocoder();
        myGeo.getLocation(point,(rs)=>{
```

```
        var addComp = rs.addressComponents;
        // Inverse resolution address
        this.position=addComp.province + " " + addComp.city + " " +
        addComp.district + " " + addComp.street + " " + addComp.streetNumber
        })
    }
```

Each alert record will generate a log. The log is automatically generated through the SLF4J tool class. The configuration information file is written in the resource folder, and the back-end automatically stores the log information with annotations. The main configuration information is as follows:

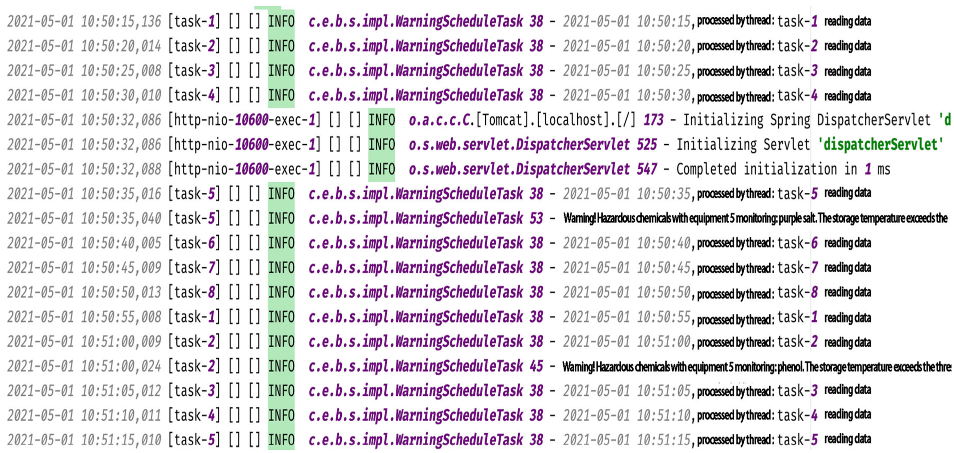
```
<configuration scan="true" scanPeriod="60 seconds" debug="false">
    <contextName>logback-warning-system</contextName>
    <springProperty scope="context" name="LOG_PATH" source="logback.path"
    defaultValue="./"/>
    <!--Output to console ConsoleAppender-->
    <appender name="consoleLog" class="ch.qos.logback.core.ConsoleAppender">
        <!--Specify log display format layout-->
        <layout class="ch.qos.logback.classic.PatternLayout">
            <pattern>[%d{HH:mm:ss:SSS}] [%t] [%p] [%X{IP}]-%C[%M:%L]-%m%
            n</pattern>
        </layout>
    </appender>
<!--Some similar logic codes are omitted here-->
<!--Specify the method class that needs to output the log and the corresponding output
level-->
    <logger name="com.example.backend" additivity="false" level="TRACE">
        <appender-ref ref="fileErrorLog"/>
        <appender-ref ref="consoleLog"/>
    </logger>
    <!--Specifies the most basic log output level-->
    <root level="INFO">
        <appender-ref ref="consoleLog"/>
        <appender-ref ref="fileErrorLog"/>
```

</root>

</configuration>

The generated log information is shown in Figure 9.

Figure 9 The log record diagram (see online version for colours)


```
2021-05-01 10:50:15,136 [task-1] [] [] INFO  c.e.b.s.impl.WarningScheduleTask 38 - 2021-05-01 10:50:15,processed by thread: task-1 reading data
2021-05-01 10:50:20,014 [task-2] [] [] INFO  c.e.b.s.impl.WarningScheduleTask 38 - 2021-05-01 10:50:20,processed by thread: task-2 reading data
2021-05-01 10:50:25,008 [task-3] [] [] INFO  c.e.b.s.impl.WarningScheduleTask 38 - 2021-05-01 10:50:25,processed by thread: task-3 reading data
2021-05-01 10:50:30,010 [task-4] [] [] INFO  c.e.b.s.impl.WarningScheduleTask 38 - 2021-05-01 10:50:30,processed by thread: task-4 reading data
2021-05-01 10:50:32,086 [http-nio-10600-exec-1] [] [] INFO  o.a.c.c.C.[Tomcat].[localhost].[/] 173 - Initializing Spring DispatcherServlet 'd
2021-05-01 10:50:32,086 [http-nio-10600-exec-1] [] [] INFO  o.s.web.servlet.DispatcherServlet 525 - Initializing Servlet 'dispatcherServlet'
2021-05-01 10:50:32,088 [http-nio-10600-exec-1] [] [] INFO  o.s.web.servlet.DispatcherServlet 547 - Completed initialization in 1 ms
2021-05-01 10:50:35,016 [task-5] [] [] INFO  c.e.b.s.impl.WarningScheduleTask 38 - 2021-05-01 10:50:35,processed by thread: task-5 reading data
2021-05-01 10:50:35,040 [task-5] [] [] INFO  c.e.b.s.impl.WarningScheduleTask 53 - Warning! Hazardous chemicals with equipment 5 monitoring: purple salt. The storage temperature exceeds the
2021-05-01 10:50:40,005 [task-6] [] [] INFO  c.e.b.s.impl.WarningScheduleTask 38 - 2021-05-01 10:50:40,processed by thread: task-6 reading data
2021-05-01 10:50:45,009 [task-7] [] [] INFO  c.e.b.s.impl.WarningScheduleTask 38 - 2021-05-01 10:50:45,processed by thread: task-7 reading data
2021-05-01 10:50:50,013 [task-8] [] [] INFO  c.e.b.s.impl.WarningScheduleTask 38 - 2021-05-01 10:50:50,processed by thread: task-8 reading data
2021-05-01 10:50:55,008 [task-1] [] [] INFO  c.e.b.s.impl.WarningScheduleTask 38 - 2021-05-01 10:50:55,processed by thread: task-1 reading data
2021-05-01 10:51:00,009 [task-2] [] [] INFO  c.e.b.s.impl.WarningScheduleTask 38 - 2021-05-01 10:51:00,processed by thread: task-2 reading data
2021-05-01 10:51:00,024 [task-2] [] [] INFO  c.e.b.s.impl.WarningScheduleTask 45 - Warning! Hazardous chemicals with equipment 5 monitoring: phenol. The storage temperature exceeds the thre
2021-05-01 10:51:05,012 [task-3] [] [] INFO  c.e.b.s.impl.WarningScheduleTask 38 - 2021-05-01 10:51:05,processed by thread: task-3 reading data
2021-05-01 10:51:10,011 [task-4] [] [] INFO  c.e.b.s.impl.WarningScheduleTask 38 - 2021-05-01 10:51:10,processed by thread: task-4 reading data
2021-05-01 10:51:15,010 [task-5] [] [] INFO  c.e.b.s.impl.WarningScheduleTask 38 - 2021-05-01 10:51:15,processed by thread: task-5 reading data
```

4 Conclusions

This paper implements a RDPSHC in energy industry based on knowledge graph. By combining the application of knowledge graph technology, the system provides the functions of management, warning and visual display for hazardous chemical risk prevention and detection. At the same time, the system also displays the system function results in the form of map information, tables and charts by means of diversified data display and monitoring charts. In the future research and learning work, we will continue to use knowledge graph and other technologies, introduce the combination of software and hardware and intelligent early warning algorithm, improve the efficiency of risk prevention and detection of hazardous chemicals and provide better decision-making and analysis services.

Acknowledgements

This work is supported by the Zhejiang Science and Technology Plan Project of China (No. 2020C03091, No. 2017C31110) and the New-Shoot Talented Man Plan Project of Zhejiang Provence, China (No. 2021R437005).

References

He, Q. et al. (2020) 'Knowledge-graph augmented word representations for named entity recognition', *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, No. 5, pp.7919–7926.

Kertkeidkachorn, N. and Ichise, R. (2017) 'An automatic knowledge graph creation framework from unstructured text', *Proceedings of National Conference of Artificial Intelligence Society JSAI2017.0*, 1N3OS39b5.

Shen, J. and Wang, Q. (2021) 'The trend of research on Alzheimer's disease based on bibliometrics and visualization using Echarts', *2021 IEEE International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI)*, pp.491–497.

Wang, Z., Tang, F. and Yu, Z.L. (2020) 'Design and implementation of a health status reporting system based on spring boot', *2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE)*, pp.453–457.

Xiaogang, J. and Xin, L. (2011) 'Research on hazardous chemicals information management system', *2011 IEEE 3rd International Conference on Communication Software and Networks*, pp.630–633.

Xu, Z., Xu, T. and Zhang, F. (2020) 'Construction of Chinese sports knowledge graph based on Neo4j', *2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology, (ICCASIT)*, pp.561–564.

Yunlong, W., Tingchun, W., Bo, M., Xiaoyan, G., Guozhi, Z. and Gujun, W. (2020) 'Risk warning for special work based on an event knowledge graph', *2020 5th IEEE International Conference on Big Data Analytics (ICBDA)*, pp.256–259.

Zhang, Q. et al. (2020) 'Construction of knowledge graph of maritime dangerous goods based on IMDG code', *The Journal of Engineering*, Vol. 2020, No. 13, pp.361–365.

Zhao, J., Hong, Z. and Shi, M. (2019) 'Analysis of disease data based on Neo4j graph database', *2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS)*, pp.381–384.

Zhou, F., Zhu, G., Yang, J. and Wu, W. (2021) 'Smart management system for hazardous chemicals park', *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, pp.659–663.