Development of Data Acquisition Software for KAERI's Radiation Monitoring System

J. Y. So [a*], S. H. Lee [b] K. P. Hong [b]
[a]*Neutron Science Center, Korea Atomic Energy Research Institute, 34166, Dajeon, Korea*
[b]*Radiation Safety Management Division, Korea Atomic Energy Research Institute, 34166, Dajeon, Korea*
[*]*Corresponding author: jiyongso@kaeri.re.kr*

1. Introduction

Traditionally, the role of radiation monitoring system (RMS) are 1) periodically collect data from various radiation detection monitors, 2) store such collected data for a long period in a safe way, and 3) visualize the data and provide related security information such as radiation alarm to interested parties. Moreover, as the social requirements of security in nuclear research area has been increased rapidly and the information technology developed, there is a need to innovate the RMS system.

In Korea atomic energy research institute (KAERI), there are many facilities in which handles radioactive materials or environmental radiation to be monitored. These facilities individually developed their own RMS systems including detectors, data management software. Therefore, as times goes on, each facilities faced the difficulties to maintain the system, especially, maintenance of software and data storage. KAERI starts the new RMS monitoring system aiming to cover the whole facilities since 2017.

In this paper, authors describe RMS monitoring software focusing on the data acquisition. Newly introduced methods are describe explicitly.

2. Methods and Results

In this section, we will summarize the current status of RMS systems in KAERI and introduce whole scheme of new RMS system in KAERI. Then the detail of data acquisition system of it will be explained.

2.1 Current Status of KAERI RMS System

Each facilities in KAERI such as Saebit fuel science facility, HANARO research reactor, 방사선 폐기물 처분장, etc. developed their own RMS systems and operated at least 10 years ago. Each RMS systems use different detectors, different software and there are no relation each other. Some RMS software are specific to certain computer or operating systems, which is now much outdated, and even impossible to operate in a currently available PC or operating systems.

2.2 Scheme and Structure of New RMS System

The first specification for the new RMS is that data acquisition from every existing radiation monitors should be possible. Not only the existing radiation monitors are now working well, but also even the replacement of radiation detectors in only one facilities requires much budget. Therefore it is an essential specification for new RMS.

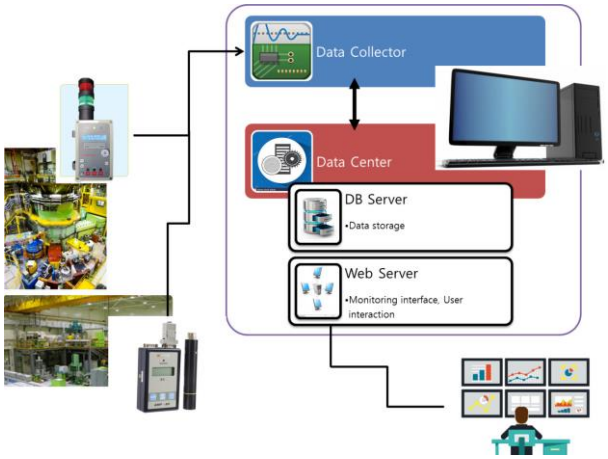

Fig. 1. The concept of new RMS system for KAERI

The second specification is that web based visualization and information providing. The RMS provides web servers and one can check the whole radiation in KAERI through web browser such as chrome, firefox, safari, explore, etc. In addition, it automatically suffices the OS independence and long-term effectiveness. The recent development of web development environments enable to various and effective visualization and information providing not only for the related personals but also for public. The other profit is that accessibility. Only if the network is connected, one can access the RMS systems.

The third specification is about the data storage. The data should be stored safely for a long time (at least 10 years) and the possibility of data loss or corruption should be eliminated. We adapt data base replication[7] for this purpose.

The forth specification is that the data acquisition part should be developed in KAERI and new types of radiation monitor should be operate by only adding the device class in data acquisition software and no additional development should be required by other part.

According to the specification we design the whole scheme of KAERI RMS system as follows. We divide the data acquisition part(we call it "collector" written in Python programming language[1]) and other parts (we call it "data center" written in C# programming

language[2]) including data base management, web server and control part. (See Fig.) A distributed messaging tool, zeromq[3] is used for the communication between the collector and data center. The zeromq provides the various communication methods among programs, computers.

2.3 Data Acquisition System

Till now, there are 7 device types in KAERI to be implemented in the data acquisition system according to their protocols. It mean that in case of different kind of device, if the company and communication protocols are same, it is regarded as the same device type

1) Mirion
2) G64
3) Dtionix2
4) Adm606
5) Victoreen94X
6) Victoreen960
7) Icam

Each of them has different connection type (RS232, RS485, Ethernet, etc.), different protocols, different kinds of measurement, different kinds of alarm and warning and different function. At first, we summarize the required functionality for the RMS systems and data center. Then we can arrange the functions to be implemented and finally can make a coherent set of functions to cover all kind of detectors and data center.

It is also very important to reduce the data acquisition time. In case of unusual occasion, one should response as soon as possible. The data acquisition time is long is equivalent to the late response. The total number of devices in one RMS system is estimated at most 100 and we decide that the total data acquisition time should not much than 20 seconds. The traditional way to collect data from many devices is using thread. However, the global interpreter lock[4] is well known that can be a obstacle for such a work. It is inevitable in python therefore we decide to use asynchronous IO[5] to overcome such a weak points. Therefore, in test phase, we check that it takes only about 6 seconds for 60 devices.

Currently the collector is running on python 3.5 and 3.6 and there is no detectible performance difference between them in case of run in same computer.

The command from data center are listed below

Table I. List of command in KAERI RMS system

initialize	Check the devices are prepared to operate. Set the data acquisition period.
addDevice	Add new device and start measurement
changeTripPoints	Change the trip points according to requested value
removeDevice	Delete the appointed device from the collector
getChannels	Return the list of device's measurement channel information.
muteDevice	Mute the alarm sound of device if the monitor device provide this function.

In the table, trip points means the assigned value in monitoring devices, and sometimes it is called as "set points". If the measured value is higher than those value, the device yields big sound for warning. Devices provide two or more values for warning.

The command and data are organized with json structure[6]. As an example, the json code for change trip points for one device are shown below. Such a structured command and response are used for the communication between collector and data center. Also the data are structured in a similar manner.

```
{
'channel': {
    'deviceId': 'a6167076-841a-44e6-b422-
dc74fa4b0588',
    'channelNumber': 2,
    'name': 'DoseRate',
    'unit': 'uSv/h',
    'collectedDate': '0001-01-01T00:00:00',
    'isActivated': "True",
    'isDisplayed': "True",
    'value': 0.0,
    'tripPoints': {
        'low': 0.0,
        'alert': 3.0,
        'high': 0.0,
        'highhigh': 25.0},
    'alarm': 'None'},
  'action': 'changeTripPoints'
}
```

The collector runs in command shall but data center has its graphic user interface(GUI) as in Fig. 2.

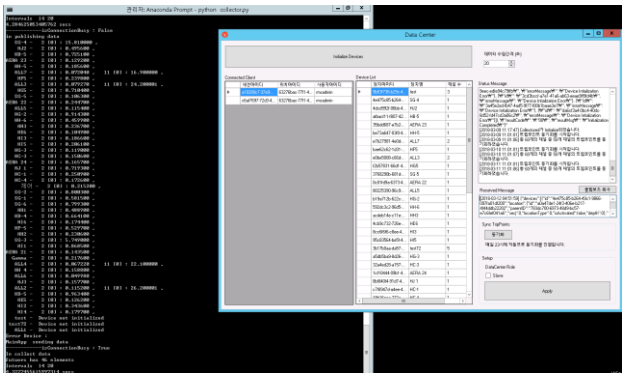

Fig. 2. collector running in the command and data center running with GUI

3. Conclusions

The new KAERI RMS systems has been developed since the early 2017 and now under commissioning. This system is developed in a new, high level programming languages such as python and c#, and the user interfaces are provided as web based form. It provides the control function for the almost device types for the radiation monitoring devices and developers make a common command interface. New technique such as message que and asynchronous IO improved the performance and developing speed. Now it is commissioning phase and we find the system is sufficient to cover the local facilities in KAERI.

REFERENCES

[1] see https://www.python.org
[2] see https://docs.microsoft.com/en-us/dotnet/csharp .
[3] see http://zeromq.org .
[4] see https://wiki.python.org/moin/GlobalInterpreterLock
[5] see https://docs.python.org/3/library/asyncio.html
[6] see http://www.json.org
[7] see https://en.wikipedia.org/wiki/Replication_(computing)