OceanSense-A general application development framework for marine environment monitoring with multi-sensors

Bin Zhao¹, Yu Yu¹*, Honghai Zhu¹, and Li Yang¹

¹ Institute of Oceanographic Instrumentation, Shandong Academy of Sciences ZheJiang Road 28, 266001, Qingdao, China

Abstract. This paper presents a component-based framework OceanSense which supports rapid development of marine environment monitoring application with multi-sensors. By decomposing systems into relatively independent, mature and reusable modules, component-based programming provides stability, reliability and extensibility compared to other programming methods. The framework covers kernel components for marine monitoring application, which are data management component, data transmission component, electronic nautical chart component, sensor communication component. OceanSense supports commonly used marine environment monitoring sensors, which increases its availability. A real project is carried on to validate the framework and result shows its availability.

Keywords. marine environment monitoring; electronic nautical chart; development framework

1 Introduction

With the rapid development of society and economy, an increasing number of human activities have gradually destroyed the marine environment. Marine environment monitoring is a vital problem and has increasingly attracted great deal of research and development attention. During the past few decades, various marine environment monitoring systems have been developed [1-6]. Based on decades of experiences on marine monitoring, a component-based framework for rapid development of marine environment monitoring application is proposed. Leveraging on large amount of related projects, commonly used sensors of different interfaces and manufacturers are summarized and catalogued. Multi-sensor supporting gives flexibility and extensibility of OceanSense.

Rest of this paper is organized as follows. The framework architecture of OceanSense is presented. Then, three kernel components of OceanSense are described. After that, an application example is presented. Finally, conclusion and the future work are given.

2 Framework architecture

As demonstrated in Fig. 1, the framework is composed by Electronic nautical chart component, data management component, data transmission component and sensor communication component. Those components can be imported into visual studio 2010. The properties of those components can be set by visual interface in visual studio 2010 and the interfaces of those components can be called by using C#. Each component will be further described as the following.

Fig. 1. System architecture

2.1 Data management component
Ocean observation data are usually sensitive data, which has high requirements for safety. A typical network architecture is described in Figure 2. Data transmission from internal network to external network can only be achieved through transmission of text files. Applications from external network periodically scan dedicated text files in the mediator system. Then, contents in those text files are compared with last time scanning contents to decide whether contents is updated by applications in the internal network. XML is preferred for its advantages in data storage and data sharing. A typical observation data in XML text files is shown in Figure 3.

**Fig. 2.** Network architecture.

**Fig. 3.** Observation in XML

### 2.2 Data transmission

Observation data transmission are usually transmitted via satellite to the ground station. In china coastal water and the Asia-Pacific region, Compass short messages provide a better way than Inmarsat for its high efficiency and low price (See Figure 5). In the near land waters, high speed data transmission with 3G, 4G or GPRS signal coverage is feasible. In the design of data communication component, communication modes are switched automatically according to different communication strategy.

**Fig. 4.** Data communication architecture

### 2.3 Electronic nautical chart component

Electronic nautical chart (ENC) component plays an important part in marine monitoring data demonstration. It provides direct connections between monitoring data and its geographical features. The extensibilities of ENC component is through supporting of IHO S-57 format. The S-57 file with .000 extensions will be translated to proprietary format, in order to ensure efficiency of S-57 objects rendering (See Fig. 6).

**Fig. 5.** Compass No1 satellites coverage area.

**Fig. 6.** ENC display.

### 2.4 Sensor communication component

Sensor communication component encapsulates the details of communications between seaMass and the instruments. A base abstract class ‘Command’ is provided to encapsulate the structure of communication command (see Fig. 7).

Based on communication direction between instrument and application, two abstract classes ‘OutCommand’ and ‘InCommand’ inherited from ‘Command’ are further given. The ‘OutCommand’ class provides ‘ToHex’ function which transfers the instantiated class object into raw hex array. Meanwhile, the ‘InCommand’ class provides ‘ValidateCheckSum’ function which calculates and validates Checksum of the instantiated class object. New command can be added by inheriting from either ‘InCommand’ or ‘OutCommand’ classes. The
Another key interface for the instrument communication component is ‘Interpreter’. The ‘translate’ function in ‘Interpreter’ interface translates the raw byte array received from serial port into concrete class object. The pseudo-code of ‘Interpreter’ is listed as follows:

1 If (buffer[2] == RealTime)
2   RealTimeCmd rtCmd = new RealTimeCmd(...);
3 else if(buffer[2] == HistoryData)
4   HistoryDataCmd hdCmd = new HistoryDataCmd(...);
5 else if(buffer[2] == SysStatus)
6   SysStatusCmd sysCmd = new SysStatusCmd(...);
7 ...

The concrete class object follows a plain ordinary java object (POJO) format. The low level details of serial port communication are encapsulated by ‘SerialPortOperation’ class, which mainly responses for serial data income event handling. The abstract classes and interface described above are the fundamentals for our automatic ‘Command’ handling mechanism. In the automatic ‘Command’ handling mechanism, ‘Command’ is expressed by XML with schemas. Taking the ‘RealTimeCommand’ for example, its corresponding XML definition is shown in Fig. 9.

Besides that, the ‘Command’ can also be validated against its corresponding XML Schema before sending out (See Fig. 10), which greatly improves system reliability.

### 4 Conclusion

In OceanSense, kernel components for marine environment monitoring application development are provided. Yet, those components with .net runtime are proved to be reliable, stable and easy to use. In future, development and refinement on those kernel components will be continued and an open source project will be founded.

### Acknowledgment

This work is supported by Shandong Academy of Sciences for Young Scientists Fund (grant no. Y15-304), Key Research and Development program of Shandong Province (grant no. 2015GSF115011), Qingdao City Southern District Science and Technology Development Fund (grant no. 2016-2-012-ZH).

### Reference


Fig.11. volunteer boat marine observation system