Development of Intelligent Interface to Input and Edit Meteorological Data

A A Shamin[1], A V Ponomareva[1], N V Kurkan[1]

[1] Tomsk Polytechnic University, 30, Lenina ave., Tomsk, 634050, Russia

E-mail: avp35@tpu.ru

Abstract. The paper presents the method of development of user interface for the hydrometeorological data acquisition system. This research includes some basic principles of creating hydrometeorological messages according to code KN-01 SYNOP. This code allows creating messages as a set of code groups. Every group keeps values of definite meteorological properties. The result of studies was implemented in creating of the user interface for the software that allows working with hydrometeorological data. The KN-01 code defines the class hierarchy of this software. The studies have shown that this method of software development is especially effective for visualization of the meteorological telegrams on devices with small displays.

1. Introduction

Climatic conditions have a massive impact on people's lives. This impact consists in not merely making daily weather forecasts, but in predicting the natural calamities and disasters. In some cases, such forecasts are time-consuming and require a continuous analysis of meteorological data, so the regular data collecting is crucially important. The weather condition observation across the globe provides a coherent view of the climatic conditions of any territory.

The hydrometeorological data acquisition system has some hierarchical structure. Each weather station reports data to a regional collecting center. Then, the regional centers deliver the data collected to the All Russian Research Institute of Hydrometeorological Information – World Data Center (RIHMI – WDC).

The hydrometeorological data are transmitted in special messages (telegrams) encoded in one of the synoptic codes. KN-01 SYNOP is one of the common codes for rapid transferring the meteorological data from the Roshydromet weather network.

Nowadays, weather stations are equipped with the systems for the automated data acquisition and transfer through a variety of channels. In automatic weather stations, one part of the telegram data is automatically generated, while another part of the message is entered manually by an operator. It is therefore relevant to develop the software to convert the data received from the sensors to a synoptic code as well as to edit these data by an operator interactively and monitor the data entry accuracy.

2. KN-01 SYNOP

Let us consider the KN-01 SYNOP scheme. This code is used to transmit data of the surface observations from fixed ground stations, which can be manned or automatic. This is a national analogue of the internationally accepted FM 12-IX SYNOP code [1].

Each section of the code consists of the code groups. A group is a set of symbols – the numbers from 0 to 9 and the «/» symbol [2]. The group length is no more than 5 symbols. With few exceptions, the length is usually equal to 5 symbols.

The sections form messages with four sections each: section 0, section 1, section 3, and section 5 [3]. The group order in a message is well-defined [4]. The weather service administration can decide on the groups which are not to be included into a weather message. Also, a group is not included if there is a parameter which was not measured in a certain period of time i.e. no information about this parameter was provided [5].

The following information is contained in section 0: the code index (in this case, the code index for KN-01 SYNOP is AAXX [5]), date and observation time, the station index number.

Section 1 includes the station type, air temperature, dew point, the current and past weather data, meteorological visibility, etc.

Section 3 includes the data about maximum ambient day/night, sunshine duration, soil surface and subsoil condition as well as other data.

Last section 5 provides a part of the weather report about the soil surface condition, air temperature, wind velocity and rainfall.

Also, some sections contain the characteristics which values define whether a certain group to be included in a report or not.

Each group starts as a rule with a certain identification number. Sections 3 and 5 start from the certain groups of the section as well [4].

An example of a message is presented below.
AAXX 01001 31878
11550 21602 11141 21161 30126 40157 53002 82030
333 21190 47005
555 1/115 3/123 51101 7990/,
where each section was written in a separate row for perception convenience.

3. Software

To check the input messages as well as to prepare and send one's own message, the relevant software and hardware should be provided. One of the facilities used for data collecting, editing and transmitting at the weather stations is the microprocessor terminal "VIP-MK" [6].

This terminal is a dedicated microcomputer with a rather restricted user interface. A standard PS/2 keyboard is used as the input device, and an LCD text with 2 rows of 40 characters – as an output device.

The limited functions of the information output at such display should be indexed when developing the user interface. The basic principles of working with similar devices are described in [7, 8] .

The main requirements to the software are as follows. As the weather data are transmitted in the form of a row which consists of code groups, it would be logical to space these groups. A user can edit some positions in the groups, but not all of them, which depends on the specific characteristics of a certain group. As discussed above, there is a set of valid characters, so that the online prompt option should be provided. That is, when moving the pointer to the next position, the information content of this prompt is changed according to this new position.

Moreover, for each code group, a short help message with a descriptive group summary is to be provided. As the display size is limited and the help message can provide quite a lot of information for online displaying, this help should be called by striking a certain key.

The data cannot be entirely displayed as the display is limited in width and height. In this case, the navigation through the line, which is longer than the display length, should be provided. It is therefore reasonable to prevent the partial displaying of code groups, i.e. when positioning the pointer directed to the row which corresponds to a new group, it should be considered that if just a part of the code group is displayed, the displayed row part should be rewritten to display this group entirely.

This software was developed using the object oriented approach [9]. Each group represents a separate class. The class objects are combined in a single vector which is practically the vector of elements with the properties of all these classes. Each group is added to the same vector of elements. To this end, every next vector element is initialized by the property values of a corresponding group. Therefore, a meteorological telegram is a vector of objects of the same type, but with different property values (with respect to a certain group). The data presented in a vector should be displayed for a user as a text string provided that the code groups are spaced for perception convenience. Considering this, a certain technique, which retrieves the string representation for each group, was used.

The class hierarchy and common logic of the application are presented in figure 1.

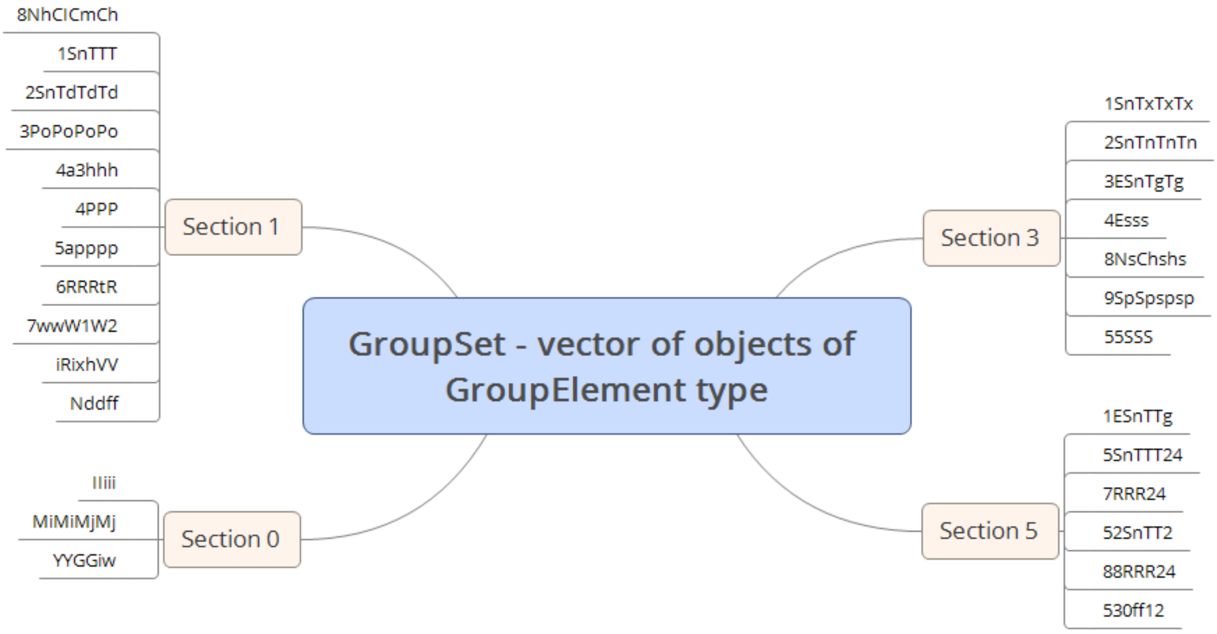
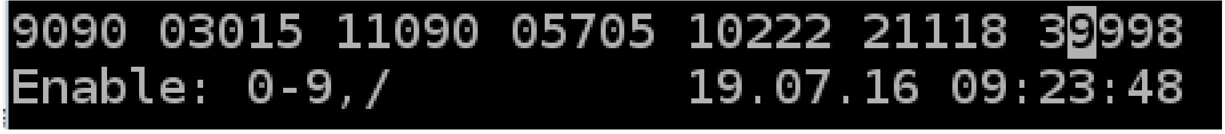


Figure 1. The class structure and meteorological telegram logic

Every class which implements a group includes its name into a group name according to KN-01. All classes are integrated into relevant sections; the initialization of the class object is section-based. The generic GroupElement class is the final vector element with properties of some part of the code corresponding to a certain group. The GroupSet class presents the telegram as a vector of the uniform elements.

The interface is presented in figure 2.



```
9090 03015 11090 05705 10222 21118 39998
Enable: 0-9,/             19.07.16 09:23:48
```

Figure 2. The example of software behavior

This kind of interface is implemented not only for input of data. It is also used in visualization of results of receiving data and data transmission. Thus, this interface is universal for all meteotelegrams. It is possible to use it for visualization of the archive of incoming messages, the archive of outgoing messages, the journal of operations.

The software discussed above was developed for using in microprocessor terminal VIP-MK based on Linux Operating System according to the main principles described in [10].

The device provides the following interfaces:

- character display 2x40 connected to VIP-MK as serial output port /dev/ttyS0 with the

speed of 192000 bps;
- the PS/2 keyboard, connected to “VIP-MK” as serial output port /dev/ttyS0 with the speed of 192000 bps;
- the Ethernet port.

“VIP-MK” is presented in figure 3:

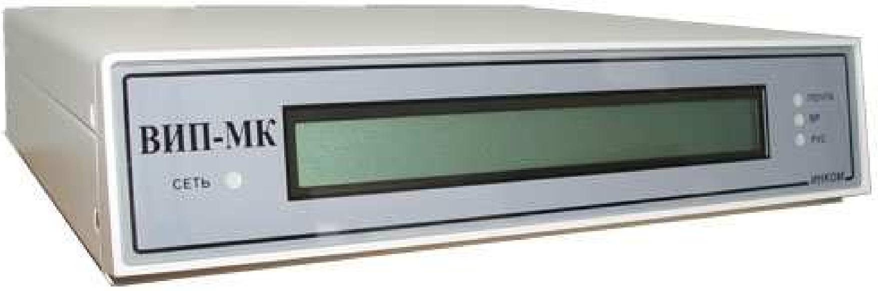


Figure 3. The appearance of “VIP-MK”

Special features of the VIP-MK terminal were considered when developing the user interface discussed above. The developed software was implemented successfully.

4. Conclusion

In this research, the algorithms and class structure in C++ were developed to implement the smart interface for input and edition of meteorological data. It was recognized that the object-oriented approach is an efficient technique to solve the similar problems. This approach enables us to develop the universal software interface as well as to modify the application when required e.g. when using another coding technique.

Therefore, the developed smart interface proved its efficiency and flexibility when processing the meteorological data.

This approach is also universal if we imply visualization of messages of different types. It means that if it is necessary to display an incoming message, an outgoing message or some operation with the message in journal, there is only one interface for it. In this case, we can write less code for displaying different types of messages. Also, it is easy to change interface. For example, if we change the coding technique or some principles of visualization, it will influence all parts of the program. This property of the interface makes the program more flexible as well.

References

[1] Vasquez T 2001 *Codes. Weather Forecasting Handbook* (Garland: Weather Graphics Technologies)

[2] Zabolotnikov G V 2010 *Study guide on the codes for meteorogical data* (St.Petersburg: PSHU Publ)

[3] World Meteorological Organisation 2011 *Manual on codes. International codes* (Geneva: World Meteorological Organisation Publ.)

[4] Russhydromet 2010 *Edict No. 372 as of December 25, Temporary instruction on the information transmission through the Poshydromet network (in Rus)* (Moscow)

[5] Russhydromet *2013 The code for the online transmission of the surface data from the Russhydromet network* (Moscow)

[6] *The Incom Group. VIP-MK: system software* (Tomsk)

[7] Sloss A, Symes D, Wright C 2004 *Arm System Developer’s Guide: Designing and Optimizing System Software* (Waltham: Morgan Kaufmann Publ)

[8] White E 2011 *Making Embedded Systems: Design Patterns for Great Software, O'Reilly Media* (Sebastopol: O'Reilly Media)

[9] Lafore R 2002 *Object-Oriented Programming in C++* (Indianapolis: Sams Publ)

[10] Yaghmour K, Masters J, Ben-Yossef G, Gerum Ph 2008 *Building Embedded Linux Systems* (Sebastopol: O'Reilly Media)