Development of Cross-Platform Software for Well Logging Data Visualization

R K Akhmadulin, A I Miraev

Industrial University of Tyumen, 38 Volodarskogo st., Tyumen, 625000, Russia

E-mail: ark@tsogu.ru

Abstract: Well logging data processing is one of the main sources of information in the oil-gas field analysis and is of great importance in the process of its development and operation. Therefore, it is important to have the software which would accurately and clearly provide the user with processed data in the form of well logs. In this work, there have been developed a software product which not only has the basic functionality for this task (loading data from .las files, well log curves display, etc.), but can be run in different operating systems and on different devices. In the article a subject field analysis and task formulation have been performed, and the software design stage has been considered. At the end of the work the resulting software product interface has been described.

1. Introduction
Processing and view of data acquired during well logging plays an important role in oil-gas production. The acquired data are essential in the development simulation model analysis and design in general as well as in the development of alternative oil production technologies [1, 2]. The information about well logging can be represented in digital form and in files of .las format.

A powerful tool in this process is the use of specialized software which can accurately and clearly provide the user with processed data in the form of well logs. Depending on a level of automation, such software can allow to solve one of the tasks of the well log analysis –identification of reservoirs containing oil, gas and water fluids. Theoretically, such software can be used under different operating systems and on different platforms [3].

This work covers the development of software for file parsing in the .las format and provision the user with necessary information about resulting well logging in the form of well log curves.

2. Research
The creation of cross-platform software for well logging visualization is an important task. It is stipulated by two factors. First, the developed software allows well logging data processing. Secondly, this product is cross-platform, which is quite efficient and profitable at the moment. A distinctive feature of this software product, apart from its cross-platform nature, is its ability to connect new plugins that can extend the functional of the product in accordance with the user’s requirements.

*.las files analysis when loading implies automatic recognition and correct downloading files of different versions. Depending on which file version has been downloaded the software will run that parser which corresponds to the current file structure. This frees the user from binding to a definite *.las file specification.

Before downloading the data files into the program their structure was analyzed. *.las files have three different standards: version 1.2, version 2.0 and version 3.0. These standards have been worked out by Canadian Well Logging Society. All the three standards have a common structure which
contains four main sections: "~Version Information", "~Well Information", "~Curve Information" and "~ASCII Log Data" (Fig. 1).

![Figure 1 – The structure of formats of *.las files](image)

The format .las version 2.0 file has been developed as a result of correcting the most common errors encountered in the format .las version 1.2 file. The .las version 3.0 file is considered the newest format, which incorporated all the required data and was created with account of the experience of versions 1.2 and 2.0. In Russia different versions of .las files are widespread.

The file section, which is read first, is "~Version Information". It contains the information about the file version. After the definition of the file version it is necessary to read the information about the well in the section of "~Well Information". This section is different for versions 1.2 and 2.0. The next section is "~Curve Information" which contains the information about the curves which in its turn includes the names of the applied techniques and their units of measurement respectively. Each well logging technique is provided with comments. This section is similar in version 1.2 and 2.0 files but has differences in version 3.0 files. The last section, that of "the~ASCII Log Data", contains a dataset in the numerical form according to the techniques described in the section of "~Curve Information". This section is also absolutely similar in versions 1.2 and 2.0, but differs in version 3.0.

The results of the files structure analysis in the .las format have become the basis of the parser class diagram. The diagram is shown in Fig. 2.

The parser architecture includes the class of "ReadersLauncher" which is the main class that reads from a file and then transmits the data stream. The classes of "LasReaderVersion" alternate a section with a version until the class corresponding to the file version switches on. Next goes reading of all the sections and data recording in the corresponding data containers. All the data containers are based on the abstract class of "KeyWDataBase". As a result of reading data from a file, all the read sections are transferred to the class of "LasData", and later to the class of "ReadersLauncher".

Non-functional requirements apart from functional ones have become an important component at the design stage of a real software product. So, the main ones are as follows:
- graphics, pleasurable for the user, and the possibility to work on the average personal computer without a high load on it;
- the support of three popular operating systems: Windows, Linux, and Mac.

Elaborate class architecture with clear partitioning of responsibility was to be designed for the software. This can be achieved with the help of object-oriented programming using the programming pattern Model-View-ViewModel (MVVM). One of the advantages of this approach is simple extendibility and the possibility of connecting plugins with the main program.
The final result of well logging data processing in the software is visualization of well logs for their subsequent analysis by the user. Additional modules (plugins) allow displaying tops and bottoms of layers as well as levels of oil and water occurrence.

Currently, when conducting geophysical surveys different-purpose software is widely used [4-6]. However, the known software products with similar functionality typically represent software systems with abundance of different functions (including the ones not required to solve the considered in the present work task) and work on a single platform – Windows. Thus, there are no analogues software with the functional similar to that considered in this work, able to operate on various platforms.

The software product design was carried out using the entity-relationship methodology to design back-end database and standardized language for object-oriented modeling UML. Use of object-oriented approach is one of the basic principles of design and has wide application in solving such problems [7-10]. Further work on the development of the program was carried out in accordance with the built at this stage models.

Since the program is designed to work with large volumes of data, it was very important to select a programming language which would be high performance. Such qualities are intrinsic to the family of programming languages C. To accomplish this work, according to the results of experiments, the C++ language was chosen. This language is widely used for the development of the various-purpose systems, including those based on complicated mathematical models and using large amounts of data, such as [11-14].

As a development environment it was necessary to choose a tool which would allow the development of a cross-platform product. In the end, it was Qt Creator – a cross-platform and open-source software development environment in QML, C++ and C.

It should be noted that Qt Creator is provided with a UI development technology Qt Quick, which allows creating your own interface in a declarative QML with minimal effort. There is also a technology of signal slots, which allows dividing the user interface and program logic into two components independent from each other, which adds flexibility in design and further modification of software.
PostgreSQL has been chosen as a database management system. This is an open-source system for database management that allows downloading unlimited amounts of data with high speed and easy-to-understand interface.

The development of the software was carried out in two stages: core development and creation of the user interface.

The core was developed with consideration of the fact that software components should be independent from each other to the maximum to ensure further extendibility without redesigning its architecture. Programming language C++ allowed achieving this independence; and the performance of the system showed a high speed under testing conditions.

The GUI was developed on the basis of QML using the technology of data transfer between a view and view model by means of signals-slots. As a result, an interface, almost independent from the software core was created, which corresponds to the task of MVVM pattern implementation.

Immediately after the main program was created the functional was extended in the form of a plug-in to display the boundaries of the tops and bottoms of layers and to display the areas of water and oil occurrence. Such data will provide a specialist with more information for further analysis of well logging data (Fig. 3). Data are downloaded from format *.csv files.

![Figure 3 – The program with the connected plugin](image-url)
As a result, the developed software has a clear interface and fully performs its task of well logging data visualization. The program can work under various operating systems without changing the initial interface and logic. The software was tested in Windows, Linux and Mac, and able to work on mobile devices, tablets, and other platforms. The developed product fully meets all the requirements, namely:

- visualizes well logging data;
- works under different platforms;
- allows to connect additional plugins.

After reading the *.las file all the well logging techniques are displayed in the main window (Fig. 4) in the form of a list. The possibility of selecting curves to be displayed on the screen has been implemented. Scaling data can be performed.

![Figure 4 – Main program window](image)

The program contains a function for combining the data of well logging interpretation and well logging (Fig. 5).
In Fig. 5, in the displayed image according to the results of combining the data, the brown color indicates the depths of oil occurrence. They all have intersecting plot lines resulting from two techniques. Thus, it can be assumed that the area with question marks can contain oil deposits.

3. Conclusion

As a result of this work, software for well logging data visualization and display of well log interpretation results has been developed. The developed software is able to display well logs and can be installed on the devices, working not only under Windows but Mac OS and Linux as well. It is also possible to run it on mobile devices under the Android operating system.

All functional and non-functional requirements formed at the design stage have been met.

The software under study is promising in terms of further development, which is achieved due to the realized possibility of new plugins development and connection to the program.

References

[1] Savinykh Y.A. Acoustic Method of Flow Rate Measurement in Multi-Layered Production / Y.A. Savinykh, H.N. Musipov // Avtomatizatsiya, Telemekhanizatsiya i Svayaz v Neftyanoy Promyshlennosti 2008. № 9. pp. 8-12.
[2] Musipov, H.N. Noiseproof Acoustic Channel in Well Drilling/ H.N. Musipov, M.A. Alexandrov, L.N. Bakanovskaya, M.V. Alferova // Avtomatizatsiya, Telemekhanizatsiya i Svayaz v Neftyanoy Promyshlennosti 2014. № 4. pp. 3-7.
[3] Akhmadulin, R.K. Cross-platform Software for Well Logging Data Visualization / R.K. Akhmadulin, L.N. Bakanovskaya, A.I. Miraev // Avtomatizatsiya, Telemekhanizatsiya i Svayaz v Neftyanoy Promyshlennosti 2017. № 1. pp. 44-48.
[4] Turenko, S.K. Adaptive Approach to Well Logging Data Processing in Building Seismic-Geological Models of Oil-Gas Facilities. / S.K. Turenko, E.A. Cherepanov // Neftyanoe Khozyaistvo 2016. № 5. pp. 34-37.
[5] Akhmadulin, R.K. The Development of the Software to Optimize Geophysical Field Oil and Gas Exploration / R.K. Akhmadulin, S.K. Turenko // IOP Conference Series: materials science and engineering. 2016. №1. Vol. 154. P. 012035. DOI: 10.1088/1757-899X/154/1/012035
[6] Dmitriev, A.N. Forward and Inverse Self-Potential Modeling: a New Approach / A.N. Dmitriev // Russian Geology and Geophysics. 2012. Vol. 53. № 6. pp. 611-622.

[7] Akhmadulin, R.K. Object-Oriented Programming When Developing Software in Geology and Geophysics / R.K. Akhmadulin, L.N. Bakanovskaya // IOP Conference Series: Earth and Environmental Science. 2017. №1. Vol. 50. P. 012049. DOI: 10.1088/1755-1315/50/1/012049

[8] Ferrari, C. LibHalfSpace: A C++ Object-Oriented Library to Study Deformation and Stress in Elastic Half-Spaces / C. Ferrari, M. Bonafe, M.E. Belardinelli // Computers and Geosciences. 2016. Vol. 96. P. 136-146. DOI: 10.1016/j.cageo.2016.08.011

[9] Zabolotin, A.E. Simulation of Triggering and Geodeformation Processes Induced by Fluid Injection Into the Geomedium / A.E. Zabolotin, A.V. Loskutov, A.V. Konovalov, S.B. Turuntaev // Russian Journal of Pacific Geology. 2016. Vol. 10, 6. P. 417-426. DOI: 10.1134/S1819714016060087

[10] Akhmadulin, R.K. An Object-Oriented Model of Case-Based Reasoning System Using Situations Tree / R.K. Akhmadulin, I.N. Glukhih, I.Y. Karyakin // Application of information and communication technologies, AICT2016. Conference proceedings: 12-14 October, 2016, Baku, Azerbaijan. IEEE, 2016. P. 124-127.

[11] Bakanovskaya, L.N. Improvement of Information Processing Quality in Preproduction Engineering Efficiency Assessment / L.N. Bakanovskaya, I.A. Chekardovskaya // IOP Conference Series: materials science and engineering. 2016. 1. 154. C. 012018. DOI: 10.1088/1757-899X/154/1/012018

[12] Glukhih, I.N. Digital Prototypes in Corporate Knowledge Bases of Well Interventions/ I.N. Glukhih, V.N. Piankov, A.R. Zabolotnev // Neftyanoe Khozyaistvo 2002. № 6. Pp. 45.

[13] Chekardovskaya, I.A. Modern Prototyping as a Basis for Industrial Design / I.A. Chekardovskaya, L.N. Bakanovskaya // IOP Conference Series: materials science and engineering. 2016. 1. 154. C. 012024. DOI: 10.1088/1757-899X/154/1/012024

[14] Bakanovskaya, L.N. Application of Artificial Neural Networks In the Heart Electrical Axis Position Conclusion Modeling / L.N. Bakanovskaya // IOP Conference Series: materials science and engineering. 2016. 1. 142. P. 012100. DOI: 10.1088/1757-899X/142/1/012100