Development of a Tool for Visualization and Analysis of 1C:Enterprise Configurations

V.V. Kopylova*, S.S. Ipantieva**, E.A. Cherkashin* ***
* Irkutsk National Research Technical University, Irkutsk, Russia
** CJSC "CROC" Irkutsk, Russia
*** V.M. Matrosov's Institute of system dynamics and control theory SB RAS, Irkutsk, Russia
kopylovika@mail.ru, saint.esp@gmail.com, eugeneai@icc.ru

Abstract - The article discusses the automation of a structural analysis and visualization of configurations of a commercial CRM software 1C: Enterprise. General scheme of the analysis and visualizing the results is based on a hierarchical clustering of configuration objects and UML class diagram representation. The software structure analysis tools allow developer to reduce time for understanding complex configurations. Examples of application and interpretation of the results are also presented.

Keywords: hierarchical cluster analysis, 1C:Enterprise, configuration structure, source code parsing, source code generation, configuration analysis tool.

I. Introduction

ERP software of Russian vendor 1C in 2017 had 31% [1]. Mostly small and medium sized companies use it. 1C applications solving a particular problem are referred to as solutions, which are configuration objects for the common interpreting platform. Companies that use 1C software have two options: either modify the software to reflect their business processes or reorganize their selves according to the standard configurations.

During its existence of the 1C platform, the number of functioning copies of configurations is constantly increasing. Russian legislation is also being improved, which requires rapid improvement of each copy of the product. Automation of enterprises, in general, requires improvements to the overall solution, making it more individual. All this contributes to the increase of objects in the configuration, which can lead to duplication of functions of some objects in others, up to full functional equivalents. Configurations consist of objects of various species: document structure descriptions, vocabularies (catalogs), indexes, reports, so called routines, etc. For each catalog and document configuration tool creates a rational table with all necessary many-to-one relations. Object structure of the catalogs and documents contains no inheritance relationships; moreover, there are no instruments to create ones. Thus, usually, the catalogs and documents have redundant structures. Implementation of an automatic analysis of the configuration on the similarity of objects allows programmer to identify duplicate objects.

From the other hand side, the prevalence of the 1C on the market of small and middle sized companies says that the configurations contain actual knowledge on structures and algorithms of data storage and processing for the standard way of common business process operations. These knowledge could be translated (transpiled) to another programming environment, e.g., for better integration with existing software.

Development of 1C software organized in a hierarchical structure [2]. Companies that automate of business processes of enterprises have developed a huge amount of code that implements various aspects of enterprise activity. The company "1C" produces a wide range of application not only for automated accounting, operational and management accounting at enterprises, but also for automation of warehousing, personal management, production management, logistics, etc. The software applications allow one to quickly generate reports, receive data, process a large amount of information. Currently "1C Accounting" application is the most popular enterprise accounting program in Russia, allowing managers to save time, optimize the number of employees engaged in accounting, as well as improve productivity.

Each configuration is a complex data structure that is interpreted within the 1C Technology platform runtime. However, a quick visual comparison of configurations from different 1C applications shows that they are developed absolutely independently of each other. In addition, within one configuration, the object structure is a non-hierarchical list of structures with common properties, which leads to the copying of structural and algorithmic information and, accordingly, makes it difficult to maintain the program code.

The structure of any 1C application configuration includes the interpreter platform, application configuration and enterprise database. Let us consider their difference in details.

Platform is the foundation of the system and the configuration development environment. The platform is called 1C:Enterprise. The main purpose of platform is to interpret configurations. The platform is developed by programmers separated from the configuration development.

Configuration (so-called "application solution") is a program developed on the 1C Enterprise platform that performs specific accounting tasks of a particular automation sector in a single enterprise. The most widely used configuration is 1C:Accounting.

Configurations have releases or revisions; the release is an update of the configuration, which is associated with the development of the application solution, the improvement of various functions, as well as changes in current legislation or market requirements. The new version of the configuration does not bring significant changes in the structure of the program from the point of view of the code.

Database is formed by the information that users enter into the system using the user interface provided by the configuration, and is stored, processed, corrected, deleted, etc.

Platform cannot perform any user tasks, as it is designed to provide configuration interpretation and perform a number of other functions. In addition, a configuration will not work without installing the platform. They only functioning as a complex system.

Another possibility of using of visualizing procedures and analysis is to support the work of a novice programmer: familiar objects are compared with others, and, to some extent, inductively it is possible to evaluate the properties of unfamiliar objects in a known way. This approach, in our opinion, will improve the productivity of novice programmers.

Although this is not typical of the 1C platform, in most software systems that support objects and object-oriented programming structures (classes) are in mutual relationships to the "inheritance" of properties. In addition, these relations can be represented as a hierarchical structure — N-ary tree. This hierarchy is natural, because professional programmers always try to introduce abstractions if there are two classes of objects that are similar in structure and properties. A common ancestor class is created, where common fields and methods of both classes are placed, thus avoiding duplication in the implementation of these elements. On the other hand, in debugged code, when creating new functions from existing classes, one can create descendant classes, where new functions are already implemented.

The aim of the study is to device software tools that allows programmer to analyze configurations, represent them in the form of UML [3] diagrams, compare the structure and properties of various configuration objects, as well as to visualize the results of this comparison.

The objectives of the study include:

1. Create procedure for importing the data structure in the format of XMI;
2. Development of object structure comparison function;
3. Construction of the system clusters using the compare function, the visualization of this structure;
4. Interpretation of the results.

II. VISUALIZATION OF THE OBJECT STRUCTURE OF 1C: ENTERPRISE CONFIGURATIONS

An UML model is an example of object models used for the design of structures of objects in a form independent from implementation in a particular programming language. UML models are stored in XMI exchange format that provides practical interoperability when working with editing tools and synthesis software code (CASE systems). In other words, UML models are documents that are used in a variety of ways, from printing images to automatically generating human-readable text descriptions of various systems [4].

A. The visualization tool

The architecture of the tools is a set of connected functional blocks:

- User interface;
- Class for processing an XML document using SAX;
- Class to generate XMI document using the document object model DOM;
- A class that implements the external connection configuration 1C.

User (programmer) loads the metadata structure of 1C configuration in XML format, which has previously been obtained using external processing, part of the delivery 1C Conversion 2.0. Then, the system analyzes the received XML file using SAX parser. The pattern of object creation starts SAXParserFactory SAXParser, which in turn via the ContentHandler interface allows one to get all the elements and attributes contained in a XML file.

Then, based on the received data, using the document object model, DOM file is generated UML model in the XMI format. In addition, programmer can use the external utilities to connect to the 1C configuration to obtain the necessary data in the TXT format. The developed software system is tested on three configurations of 1C, which are the most popular in the small and medium business:

- Retail 2.0 (managed forms, typical configuration),
- Accounting enterprise 2.0 (normal application, typical configuration),
- Management of funeral services 1.5 (managed forms, industry solution).

Test process resulted in the XMI file generated and imported in the Modelio Open Source environment, for example a part of Class diagrams for the document "the retail sales Report" shown in Figure 1.

Based on data from SAX-analyzer [5], an UML model was formed in the XMI format. To do this, we use the document object model DOM because DOM is used to parse XML documents and construction with a standard API, unlike the SAX parsers, which are used exclusively for XML parsing.

The document object model DOM represents XML files (and therefore XMI files) as trees of nodes. XML parsers allow us to write the DOM tree in the XMI file and get the DOM tree from the XMI file.

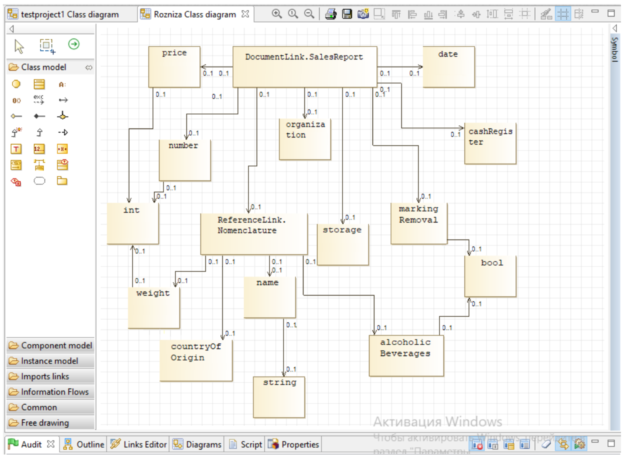

Figure 1. Class diagram for document "Retail sales report".

B. An object structure analysis technique for 1C: Enterprise configurations

In the analysis, we assume that the objects of 1C configuration correspond to some hierarchical structure of inheritance, and these objects are the only instances of these classes. Thus, under the similarity of the two objects, we will further understand the properties of the location of the corresponding classes in this hierarchy. Objects are more similar to each other if there are fewer structural elements (vertices and arcs) between their classes along the path connecting these classes.

The results of hierarchical clustering have a similar property. To build such a clustering, you need to create a two-argument function that produces 0 if both arguments are identical (in particular, the same), and some positive number if the objects do not match, and the more objects "do not match", so this number will be greater. Such a function measures the dissimilarity of two objects.

C. Hierarchical clustering

Hierarchical clustering is a set of algorithms organizing data in order to create a tree of nested clusters. There are two classes of hierarchical clustering methods:

- Agglomerative (connecting) methods: new clusters are created by combining smaller clusters and thus the tree is created from leaves to trunk;
- Divisive (separating) methods: new clusters are created by dividing larger clusters into smaller ones and thus the tree is created from trunk to leaves.

In this report, as a measure of the dissimilarity, we will use general characteristics of the structure of the object: the number of fields (or attributes), the number of table parts and the number of fields of these table parts (attributes in tables). These parameters are the three coordinates in the Euclidean object space. Source data about object structures is loaded from XML-file. The dendrogram is constructed by agglomerative method, first combining the most similar objects into sets. Root is the top of the dendrogram is, in fact, is configuration.

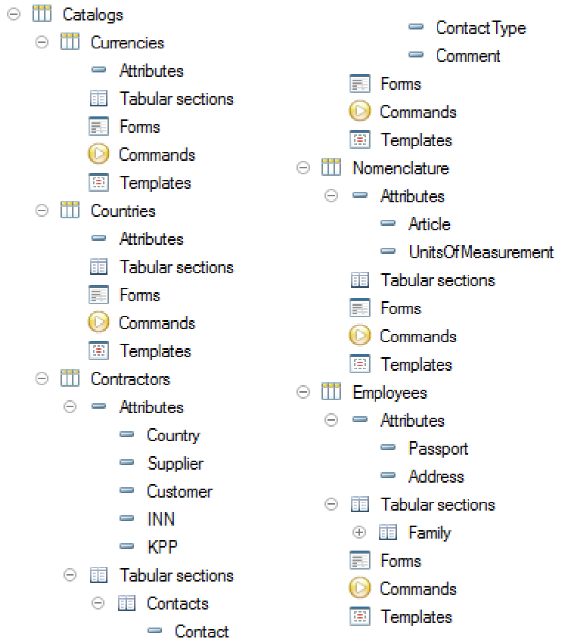

Figure 2. Composition of the test configuration

As an example, consider a simple configuration specifically designed to test the proposed technique. The configuration consists of five directories: Currencies, Countries, Contractors, Nomenclature and Employees (figure 2).

Directories of Currencies and Countries consist of fields "Mark Removal", "Code" and "Name". Details of the directory Nomenclature contains more fields: "Mark Removal", "Code", "Name", "Article" and "Units of measurement". The structure of the directory Contractors and Employees include also a tabular part. Contractors details are "Mark Removal", "Code" "Name", "Country", "Supplier", "Customer", "INN", "KPP"; table part is "Contacts" consisting of details "Contact", "Contact Type", "Comment". Employees comprises details "Mark Removal", "Code" "Name", "Passport" and "Address"; table part is "Family" consisting of details "Role", "Full name".

The reading process of the structures of the objects in the XML configuration file is a sequential tree traversal in depth. In the analysis state, we use a method of processing the source file that integrates DOM with the object properties of the Python — ElementTree. The ElementTree library represents the DOM as language objects [6] rather than as low-level abstract library objects, which greatly simplifies the processing of the original XML data.

Since the purpose of reading the XML configuration document is to collect generalized data about the structure of objects, when the desired element of the structure is found, data about it is recorded in the appropriate table of the local database implemented using the library SQLite 3 [7].

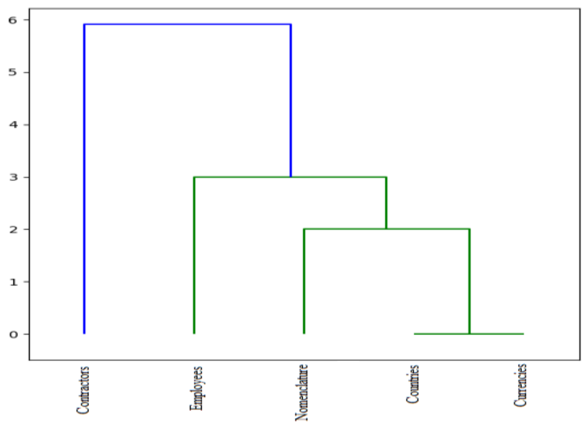

Figure 3. Dendrogram for test example

Recall, that the number of fields, the number of table parts, as well as the number of fields in the table parts, which gives the Euclidean space with three coordinates, characterize each configuration object. For the objects presented in this example, the coordinates are shown in Table 1.

TABLE I. EXAMPLES OF COORDINATES OF OBJECT FEATURES

Reference	*Coordinates*
Currencies	[3,0,0]
Countries	[3,0,0]
Contractors	[8,1,3]
Nomenclature	[5,0,0]
Employees	[5,1,2]

Example of calculation of the distances between objects are in Table 2, for these distances dendrogram have been built (Figure 3) with clustering.

TABLE II. DISTANCE BETWEEN OBJECTS (A MATRIX OF DISSIMILARITIES)

	Currency	**Countries**	**Contractors**	**Nomenclature**	**Employees**
Currencies	0,0	0,0	5,916	2,0	3,0
Countries	0,0	0,0	5,916	2,0	3,0
Contractors	5,916	5,916	0,0	4,395	3,162
Nomenclature	2,0	2,0	4,395	0,0	2,236
Employees	3,0	3,0	3,162	2,236	0,0

Since all the data on the structure of the objects stored in a database, the structure of the object easily retrievable and can be represented in a formal language.

```
Reference
ReferenceLink.Currencies
Props MarkingRemoval
Props Code
Props Name
/Reference
Reference
ReferenceLink.Countries
Props MarkingRemoval
Props Code
Props Name
/Reference
Reference   ReferenceLink.
Contractors
Props MarkingRemoval
Props Code
Props Name
Props Country
Props Supplier
Props Customer
Props INN
Props KPP
TablePart Contacts
Props Contact
Props ContactType
Props Comment
/TablePart
/Reference
Reference
ReferenceLink.
Nomenclature
Props MarkingRemoval
Props Code
Props Name
Props Article
Props
UnitsOfMeasurement
/Reference
Reference
ReferenceLink.
Employees
Props MarkingRemoval
Props Code
Props Name
Props Passport
Props Address
TablePart Family
Props Role
Props FullName
/TablePart
/Reference
```

Similarly, one can get a representation of the data structure of an object in another object-oriented language, for example in MQL4. MetaQuotes Language 4 (MQL4) is a built-in programming language for trading strategies developed by MetaQuotes Software Corp. based on its many years of experience in the creation of trade and information platforms. This language allows programmer to write Expert Advisors programs that automate the management of trading processes and are ideal for implementing own trading strategies. Besides, using MQL4, one can create your own custom indicators, scripts and libraries.

III. RESULTS AND DISCUSSION

Consider two identical according to the given dissimilarity function structures of the directories "Currencies" and "Country". They currently contain no fields or table parts. Both directories have standard attributes "Mark Removal", "Code" and "Name". Then one can create a class "Base" and fill it with the last three standard attributes.

```
Class Base {
    protected:
        bool MarkRemoval
        int Code
        string Name
}
```

Our directories will be the children of the created Base class.

```
Class Currencies : public Base { }
Class Countries : public Base { }
```

Continuing building classes based on inheritance, we get the following code for rest of structural objects.

```
Class Nomenclature : public Base {
    private:
        string Article
        string UnitsOfMeasurement
}
Class Countries : public Base {
    private:
        string Passport
        string Address
        table Family {
            string Role
```

```
            string Name
        }
}
Class Countries : public Base {
    private:
        Country aCountry
        bool Provider
        bool Customer
        string INN
        string KPP
        table Contacts{
            string Contact
            string TypeContact
            string Comment
        }
}
```

IV. Conclusion

We developed a software system for the visualization of the configuration structure 1C allows programmer to analyze the metadata structure of 1C configurations by conversion in an UML model in the XMI format. Visualization method discussed in the study implements the international standards for the presentation of configuration objects of an application, in contrast to available on the current market.

In the report, the technique of the generalized analysis of structure of objects of configuration 1C is offered. The technique is based on the construction of a dendrogram with an agglomerative method based on the "similarity" of object structures. The quantitative characteristics of the structure of objects are used as a measure of similarity. An example of the test configuration analysis of five reference objects is given.

Using the developed methodology and tools for analyzing 1C configurations allows one to get a visual representation of the structure of 1C:Enterprise configurations in the form of hierarchies of objects. It is useful to localize duplication of objects having the same structure, as well as to obtain a program for representing objects in another object-oriented language that supports inheritance of properties.

Future activity of the project should be organized in the following directions:

1. Improvement of the dissimilarity function to optionally account type and names of attributes;
2. Development of a tramspiler, a translator of 1C object structure to another high level language;
3. Device a refactoring tool for joining common object code in a so-called common module.

V. Acknowlegement

The work is supported by the Russian Foundation for basic research, grants 17-07-01341, 18-07-00758, 17-47-380007.

References

[1] Market share ERP systems In Russia [Electronic resource]. URL: http://www.tadviser.ru/index.php/ERP. (in Russian) (Date of access 15.05.19)

[2] 1C:Enterprise 8. Program system – [Electronic resource]. URL: http://v8.1c.ru/predpriyatie/questions_licence.htm (Date of treatment 15.05.19)

[3] Novikov F. A., Ivanov D. Yu. a. Modeling in UML. Theory, practice, video. // SPb.: Professional literature, Science and technology, 2010. — 640 pages

[4] National open University INTUIT – [Electronic resource]. URL: https://www.intuit.ru/studies/professional_skill_improvements/1210/courses/6/lecture/182?page=2 (Date of access 15.05.19)

[5] IBM Developer – [Electronic resource]. URL: https://www.ibm.com/developerworks/ru/library/x-xjavaforum5/index.html (Date of access 15.05.19)

[6] The Community of Programmers. The source code of our knowledge - [Electronic resource]. URL: https://python-scripts.com/object-oriented-programming-in-python (Date of access 15.05.19)

[7] The Community of Programmers. The source code of our knowledge - [Electronic resource]. URL: https://python-scripts.com/sqlite (Date of access 15.05.19)