A software platform for constructing model-driven systems for primary data consolidation

Aleksei A Korobko and Anna V Korobko
Department of Applied Computer Science, Institute of computational modelling SB RAS, Krasnoyarsk, Russia

E-mail: gglhroom@gmail.com

Abstract. The effectiveness and validity of decisions in all subject areas depend on the completeness and reliability of the information base. Despite the development of information technology, some part of data is collected manually. This is important for the tasks of updating normative references, recording results of observations and laboratory studies, and statistical reporting. The paper describes the authors’ approach to building typical model-driven data collection systems. The approach includes the original conceptual model, the order of building systems by forming a control model and a set of algorithms for the construction of the system. Theoretical results form the basis of a software platform for building typical model-driven systems for collecting and consolidating primary data. The paper describes the procedure of constructing a data collection system on the basis of the platform using the example of the Soil Condition Research System. The software platform provides the systematization of consolidated data and their consistent storage. In addition, the control model of the system is a formal description of its subject area and may be useful for exploring the data collected.

1. Introduction

The effectiveness and validity of decisions in any subject area depend on the completeness and reliability of the information base. Despite the development of information technology, a significant part of data is collected manually. The collection of primary information is relevant for various areas, for example, to monitor the results of activities of scientific, medical and other institutions, to monitor and predict emergencies, to conduct research and to solve many other applied problems related to data collection.

Different areas have different requirements for the process of collecting information. Emergency management requires quick response and tactical decisions. The efficiency of obtaining primary data is important here. Often the composition of the collected information is changed during the collection of primary data. The data structure is changed when the law is changed, scientific experiments are expanded etc. The requirements for operational consolidation and dynamic development of the primary data consolidation system are essential.

In most cases, at the initial stage of automation, tabular editors are used for data collection [1, 2]. Over time, there is more understanding of the process of data collection, awareness of the need to unify the process of data collection, and centralization of the database. Based on the requirements established, a highly specialized system of data collection is created. An external organization may develop a collection system. Then, the development of the system is very slow, because for each iteration of the development it is necessary to prepare a technical task and allocate money. Or, there is a Development department within an organization. Then, the system is built within this organization and for its support
and development its own specialists are attracted. Depending on the requirements, the development of the system may be more intensive. In any case, the system must evolve. Without proper maintenance, the data collected by the system become irrelevant and incomplete, the system ceases to help employees in their work, becomes a burden and at a slightest opportunity, employees cease to use it. Moreover, it becomes impossible to make decisions based on the collected data, as their reliability and consistency are questioned.

One of the approaches to accelerate the development of the system is the use of models. Typically, models are used to document and coordinate analysis, design, and implementation tasks [3]. With the development of information technology, there appear new opportunities to improve the technology of data collection and development of new approaches to the creation of primary data consolidation systems. One of such approaches is the Model-driven development (MDD). MDD uses models as a Central component of the software development process, all models are divided into levels of abstraction, the transition from one level of models to another is regulated, and the result of the development is the code of the system (https://www.omg.org/cgi-bin/doc?ormsc/14-06-01.pdf). The code refers to the description of the database structure, source code of classes in the target programming language, etc.

The model-driven approach is mostly aimed at automating the development process. The development of the resulting system by a user is difficult because for this purpose it is necessary to master the tools of building models, understand various notations of models and learn how to build the resulting code into the working system.

This paper proposes an original approach to the construction of model-driven systems, which consists in the formation of the thematic core (control model) of the system using the built-in interface [4]. A set of algorithms for automatic generation of application models and the system interface based on the control model allows one to instantly change the content of the collected information. This allows model-driven systems to be developed by users without special knowledge [5, 6].

The analysis of the data collection process and the authors' experience in the implementation of consolidation systems in different subject areas allowed one to develop a single conceptual model of data collection. This model, in conjunction with the authors' approach, is the basis of a software platform for building model-driven systems for the live data collection and consolidation of primary data.

The purpose of this study is to describe the theoretical foundations and implementation of the software platform which uses the original approach and provides the construction and development of model-driven systems for the live data collection and consolidation of primary data in various application areas.

Here, the authors' approach to the construction of model-driven systems is presented in comparison with the existing software design tools based on the MDD approach. Formally, we describe a universal conceptual model for the construction of standard data consolidation systems. A software platform for building typical model-driven systems for collecting and consolidating primary data is presented. The functional capabilities of the proposed platform are illustrated by the example of the problem of assessing the degree of soil degradation, and anthropogenic and technogenic load on soils using the bioluminescent enzymatic analysis.

2. Overview of MDD based software design tools
There are numerous tools to facilitate the construction of models within the MDD approach. Some of them are universal. For example, the StarUML, a tool based on Unified Modeling Language (UML), allows one to build different models. Transitions between models can be performed either using the tool-supported plug-in system or manually.

In addition, there are specialized software tools that allow automating some processes of model-driven development, for example, Rational Software Architect, Simulink, Sirius.

IBM Rational Software Architect is a modeling and development tool which uses UML to create applications and web services in C++ and Java.
Simulink created by MathWorks is a software environment for simulation. With this tool, users can build models, test, and automatically generate codes. Simulink Coder is used to automatically generate codes based on models.

Sirius is an open-source graphical modeling tool created under the auspices of the Eclipse Foundation. Sirius allows one to create models using charts, trees, and tables. The user does not need advanced technical knowledge to create models because the control model is described in the domain language. Also, one does not have to work with the source code. Sirius is used for designing complex systems, for example, for programming robot behavior using special blocks-commands which are understandable to operators.

As can be seen from the review, the existing tools perfectly serve to automate the software development process, including that within the framework of the MDD technology. However, in order to use universal tools, it is necessary to understand the model notations, as well as the MDD technology itself. Specialized tools are created on the basis of various programming environments, the result of using these tools being the source code or intermediate format of models intended for further processing. To use such tools is difficult for untrained specialists.

The development of methods and means for the creation of data collection systems based on the model-driven approach developed by the user without the involvement of programmers remains relevant.

3. An original approach to the construction of model-driven systems

Model-driven development (MDD) technology is based on the construction of an abstract metamodel, control and application models as well as on specifying ways to transform models into supported programming technologies. The construction of model-oriented systems within the MDD approach includes the stages of designing abstract-level models and application-level models [7]. The design of abstract-level models involves the process of constructing a meta-meta model (model of the modeling language) and the process of constructing metamodels for various specific application areas. The design of applied models includes the process of building models of the application level and the stage of forming instances of concepts defined at the abstract level [8].

In order to create data collection systems based on the model-driven approach, the author proposes a modification of the classical MDD process. Firstly, the processes of forming the control and applied models are combined. The combined process assumes that the control model of the system is formed by the user using the interface of the system itself. This automatically changes the structure of the database and preserves the formal description of the models. Secondly, the process of automatic transformation of the applied models into the user interface of the model-driven system is added. Due to this, the system interface is automatically rebuilt in terms of the data input, in accordance with the changes made by the user in the control model.

To solve the problem of the development of typical model-driven data consolidation systems, a meta-model of data collection is proposed, which is a specification for the creation of the control and applied models of the designed systems [9]. The meta-metamodel includes three classes of objects: class "Object" – N, class "Attribute" – F and class "Group" – G. Objects of the "Attribute" class are described by the triple F = (A, T, D), where A is the name of the attribute, T stands for the name of the specialized attribute type, and D for the attribute temporality. Within the meta-metamodel, a number of relationships are defined which determine the relations between the instances (objects) of the model classes. There are two types of relations between objects, "Nesting" and "Dependency". One-multiple-valued relation "Nesting" – ϕ, is defined on the set N, ϕ⊆N × N, being intended for setting the organizational hierarchy of objects. The one-to-many relation "Dependency", which we denote as χ, is given on the set N, χ⊆N × N. The relationship allows correlating objects with each other, realizing various functional communications. The multi-valued correspondence between objects and attributes is "Ownership", denoted as θ, where θ⊆N × F. One-to-many-valued correspondence between objects and groups, i.e. "Consolidation", is denoted as ψ, where ψ⊆N × G.
To test the proposed theoretical approach, conceptual model and transformation algorithms, a software platform for building typical model-driven data collection systems has been developed.

4. Building model-driven systems based on the platform

Data collection systems created based on the platform use relational databases. In the database, the system stores service information, the control model, as well as data collected during the main course of operation.

Building a new system based on the software platform begins with the creation of a database in Postgres by the DBMS server administrator. The database is created using a template. The template contains basic structures for storing model-driven system metadata. The access settings to the future system are filled with data allowing the first user, its administrator, to log in. The administrator has the right to add new users and configure the system. The data collection system uses the web interface to operate. Therefore, a new domain is registered for the system within the framework of the platform, and the database connection parameters are defined in the settings of the web interface. As a result, the administrator has access to the system through the Internet.

The system interface consists of three main modules: operator, moderator, and analyst. Each module is accessed according to the user role.

In the first step, after creating the system, the administrator creates users in the system and distributes the access rights. The system allows distributing access rights according to user roles.

After creating users, the work of moderators starts. The work of users having the role of the moderator is crucial and requires good knowledge of the subject area, moderation tools and rules of building the system structure. The moderators are engaged in the development of the system. The moderator module is divided into two parts, one part allows creating attributes F, while the other part allows creating objects N (figure 1). The attributes interface allows the moderator to create simple attributes and those based on χ classes, customize their properties, and group them.

![Class editor with class description Mobile nitrogen and phosphorus in soil](image)

**Figure 1.** Class editor with class description Mobile nitrogen and phosphorus in soil.

The class creation interface allows the moderator to add the necessary attributes to the class and set the main and additional properties of the class. Attributes can be used in different classes. Their change
is reflected in all the classes associated with the attribute, ensuring consistency of the same indicators throughout the database. The classes, attributes, and relationships created by the moderator form the control model and define the structure of the automatically generated database. Based on the class properties and associated attributes, primary data entry forms are dynamically created in the operator Module.

Once the moderator has configured the system, operators begin to enter data. The operator module allows one to select the required form. After selecting the form, there appear the section containing the previously entered data and data editing section (figure 2). The interface of the operator module is formed based on the control model. If the moderator modifies the control model, the data entry interface will be automatically rebuilt the next time it is updated. Thus, the development of the system occurs with the minimum delays between the modeling stage and the stage of rebuilding the system.

Analytical reports are created as database views by the DBMS administrator. The system automatically identifies the view as a report by the prefix in the title. The report description and field names are specified as view descriptions. The analysis module displays a list of report descriptions that are available for analysis. After selecting a specific report, one can analyze the data using one of the following tools: table, line chart, scatter chart with K-means clustering, and cartogram. For deeper analysis, the report data can be transferred to the third-party Analytics systems such as PowerBI and Tableau.

The developed software platform allows model-driven systems to be created and developed by users, without involving programmers.

5. Example of building a system
The development of model-driven data collection systems is particularly relevant for solving the problems of data consolidation in scientific research. For scientific activity it is important to store retrospective data, ensure the integrity and consistency of information, provide opportunities for collective work with data, as well as to take into consideration the specific characteristics of the studied objects and possibility of expanding the scope of research.
Regardless of the complexity of the experiments and field of knowledge, all studies are carried out according to a similar algorithm, which includes the following processes: observation, hypothesis building, experiment, analysis of the information obtained and forming conclusions [10]. Ideally, the data collection system should accompany all the stages of the study. Let us consider the system of research of soil cover condition which is created based on the software platform for model-driven data collection systems [11].

Here, the object of the study is the soil cover and its ecological state. To assess the state of soils, various objects characterized by territorial binding are studied.

Currently, the system of soil cover research contains information on several experiments. The main hypothesis of the research is the possibility of using bioluminescent enzymatic analysis as a method of biotesting to assess the degree of soil degradation, anthropogenic and technogenic load on soils. To confirm the hypothesis, changes in the value of residual luminescence (T) in the soil profile with increasing depth of sampling and with changes in soil properties (humus content (%), pH of aqueous and salt suspensions, nitrate-nitrogen content (mg/kg), mobile phosphorus (mg/100g)) are studied.

Initially, the system included the results of field studies, namely the assessment of the ecological state of the soil of the settlements of the Krasnoyarsk Region: Minino-Pogorelsky Bor, Solyanskoje, Sayano-Pogranichnoye. At each of the objects of research full-profile soil sections were established, binding and description of soils according to the morphological features were performed, the full name of the soil was given, and soil samples were selected at different depths.

According to the specification of the control models of model-driven systems for working with the object under study, a class “Sampling Place” was created in the Moderator module, which was associated with its characteristic attributes (Name and Description). To create an application model when creating a class and attributes, a few utility properties (properties related to the database and interface settings) were indicated. At each sampling site, several soil sections could be made, which were described based on the substantive genetic approach [12]. Analysis of the documentation for the experiment made it possible to identify additional attributes of the soil section: Section code, Coordinates, Soil type, and Conditions. In the system, the Soil section class was created, which was associated with the classes “Site of sampling” and “Type of soil” and with the selected attributes. The relationships between the classes in the system data input module were implemented in the form of lists, ensuring the integrity and consistency of the collected data.

The soil samples were taken in different horizons of the section having the following attributes: horizon index, minimum layer depth, maximum layer depth, and power. The class “Section horizon” was created in the system, which was associated with the classes “Soil section” and “Horizon index”. Relations with the attributes characterizing the horizon of the section were also established.

To store information on the chemical composition of natural samples, the class of the same name was created. This class was associated with the “Section horizon” class, allowing one to restore complete information on the location of the experiment, section and section layer from which the analyzed soil sample had been taken. The class was also associated with the attributes: "Humus", containing information on the percentage of humus in the sample; "Exchange acidity" (pH salt); “Actual soil response” (pHH2O); "The content of nitrate-nitrogen (or mobile nitrogen)” (N-NO3); "The content of mobile phosphorus" (P2O5). The results of biotesting of the natural samples were stored in the class of the same name with the attributes: “The value of the residual glow” (T) and “Standard deviation”.

Then, to study the dependence of the results of biotesting on the duration of storage of soil samples and conditions of sample preparation performed by a group of students, it became necessary to develop the class “Biotesting of natural samples”. The attributes “Date of investigation” and “Type of sample preparation” were added to the class, forming an independent class. The values of the attributes of the “Type of sample preparation” class describe the sample preparation options for the study, including various heat treatment options and their combinations.

To continue the study of natural samples, sampling and analysis of technogenic-surface formations were performed near the industrial enterprises of the city of Krasnoyarsk. The samples were analyzed for the concentration of lead, arsenic, and fluorine. Agrochemical indicators were studied. Accordingly,
the control model of the system was supplemented by the classes: “Lead Test”, “Fluorine Research”, “Arsenic Research” and “Agrochemical Indicators”. The new classes were associated with the class “Sampling point” since the sampling for the study was carried out from the surface and did not involve soil sections.

The Krasnoyarsk reference center conducted an experiment with reference soils to calibrate and determine the boundaries of the effectiveness of bioluminescent enzymatic analysis as a biotesting method to assess the degree of soil degradation, as well as to estimate anthropogenic and technogenic pressure on the soils. The studies were carried out using various enzyme systems using standard methods of reference samples for common pollutants. In a separate section “Laboratory research”, a number of classes were created for storing the results of the study: class “Analysis for pesticides”, class “Analysis for copper”, class “Physicochemical properties of the sample” and class “Biotesting of reference samples”. The class “Physico-chemical properties of the sample” was associated with the class “Variety in particle size distribution” and with the class “Reference soil sample”. The creation of the “Reference soil sample” class was due to the fact that it significantly differed from the natural ones. In particular, the reference samples did not have any sampling site, and hence, no geographical reference.

Thus, at present, the "System of soil cover research" contains the data from five different experiments. Due to the fact that the system is built on the basis of the proposed software platform, the experimental data are in a consistent state. The consistency and completeness of the data allow one to perform a comparative analysis of the results of the related experiments and to analyze the data of long-term experiments. The possibilities of the platform and the proposed authors' approach allow the researchers to develop the information system in accordance with the progress of scientific research.

6. Conclusion

The completeness and reliability of the information used for decision-making determines the degree of efficiency and validity of management decisions in any subject area. In the context of widespread automation of data receipt and consolidation, a significant part of the data is still collected manually. This concerns the tasks of updating reference books based on normative and legal information, recording the results of observations and laboratory studies and presenting statistical reports.

The experience of developing several data collection and storage model-driven systems over the past five years and the systematization of the accumulated knowledge has allowed the authors to develop a universal meta-model of the data consolidation process. The meta-metamodel describes the main classes of entities and relationships between them. They are involved in the process of data collection and used to build control models of specific application systems. The original approach to the construction of model-driven systems and a set of algorithms for automatic generation of applied models and system interface based on the control model has formed the theoretical basis of the platform for building data consolidation systems.

On the one hand, the platform confirms the reliability of the theoretical results, and on the other hand, it solves the urgent application problem considering modern requirements for software products. Based on the platform the following is built: a data collection system to account for the results of the publication activity of researchers, a subsystem for preventing and mitigating the consequences of emergency situations and a system for recording the results of field and laboratory studies.

The paper describes the authors’ approach to the construction of model-driven systems and a universal conceptual model that provides the construction of typical data collection systems. A software platform for building typical model-driven systems for collecting and consolidating primary data is presented. It describes how to build a data acquisition system based on the platform and gives an example of the creation and development of "System studies of the condition of the soil cover".

The use of the software platform for building data consolidation systems provides the systematization of the consolidated data and their consistent storage. In addition, the control model formed during the construction and development of the system is essentially a description of the subject area and, by analogy with the models of knowledge representation, can be reused. The actual task for the further
research is to build an analytical model based on the control model and to support the visual design of research queries to the accumulated data.

Acknowledgments
The study was carried out with the financial support of RFBR and the Government of Krasnoyarsk region, research project №18-47-240005.

References
[1] Broman K W and Woo K H 2018 Am. Stat. 72.1 2-10
[2] Panko R R 2015 Shidler Coll. Bus. 10.2 15-21
[3] Grace P, Truyen E, Lagaisse B and Joosen W 2007 Proceedings of the 6th International Workshop on Adaptive and Reflective Middleware 2
[4] Korobko A A and Nozhenkova L F 2016 Proc. 39th International Convention The conference ”MIPRO 2016” 1729-34
[5] Korobko A A 2016 Educational resources and technologies 2(14) 173-80
[6] Korobko A 2018 IEEE 3rd Russian-Pacific Conference on Computer Technology and Applications (RPC) 1-4
[7] Seidewitz E 2003 IEEE Softw. 20.5 26-32
[8] Atkinson C and Kühne T 2003 IEEE Softw. 20.5 36-41
[9] Korobko Aleksei A and Korobko Anna V 2015 Modeling of Artificial Intelligence 6.2 82-9
[10] Lynch S M 2013 Using Statistics in Social Research: A Concise Approach 1 1-15
[11] Kratasyuk V and Esimbekova E 2015 Chem. High Throughput Screen. 18.10 952-9
[12] Shishov L, Tonkonogov V, Lebedeva I and Gerasimova M 2004 Classification and diagnostics of Russian soils (Smolensk: Oikumena) p 56