

A New Approach for Software Development in Terms of Problem-Oriented Knowledge Search and Processing

Andrey Lezhebokov, Bogdan Shkalenko and Elmar Kuliev

Abstract The paper deals with a development of a new approach of knowledge representation, which can be used in design the software to support of search and processing of problem-oriented data and knowledge. The authors formulate fundamental principles of a object-oriented knowledge repository building. Also there is a software architecture on the basis of object approach and object-oriented knowledge base, which describe knowledge in a subject area with maximum precision. The paper describes current criteria of estimation modern software development systems, presents the analytical review of existing programming aids and techniques. To design modern easy-to-use software environment the authors determine the most appropriate development tools. On the basis of chosen development tools we implement a cross-platform software and describe users' interface. Further research is directed to determine the possibilities of high-level flexible frameworks to create modern cross-platform programming solutions.

Keywords Software · Information system · Knowledge search · Problem-oriented knowledge · Object approach · Database · Knowledge base

1 Introduction

At the present time a lot of companies and agencies rapid accumulate huge amounts of data (BigData) in different fields. There are plenty of BigData sources in modern world. For example, time-line data from measuring devices, events from radiofrequency identifier, social network message flows, meteorological data, rEarth Observation (EO) constellation, data flows from mobile cellular about subscribers

A. Lezhebokov · B. Shkalenko · E. Kuliev (✉)
Southern Federal University, Rostov-on-Don, Russia
e-mail: elmar_2005@mail.ru

A. Lezhebokov
e-mail: legebokov@gmail.com

location, devices of audio and video registration [1, 2]. Mass distribution of technologies and new models of different devices and internet-services became a starting point for BigData entry into all spheres of human life. Primarily, there are scientific research, business sector and government.

Accumulated information increased so much that there is need to develop modern approaches, principles, technological and practical tools for its processing and analysis. Stored data gives an opportunity to extract new knowledge. With the use of this knowledge we can obtain analytical information and control the companies' development strategy, predict market trends and find new optimal solutions, which provide a successful development in a competitive climate. In modern world such information analysis is essential part of daily activity for many organizations [3, 4].

Today, an optimal data storage problem is the most promising. It allows to extract critical information from accumulated data most effectively. This paper studies the representation method and data base management system (DBMS) capable to approximate the data storage structure to the real subject area.

2 A New Approach for Problem-Oriented Knowledge Representation

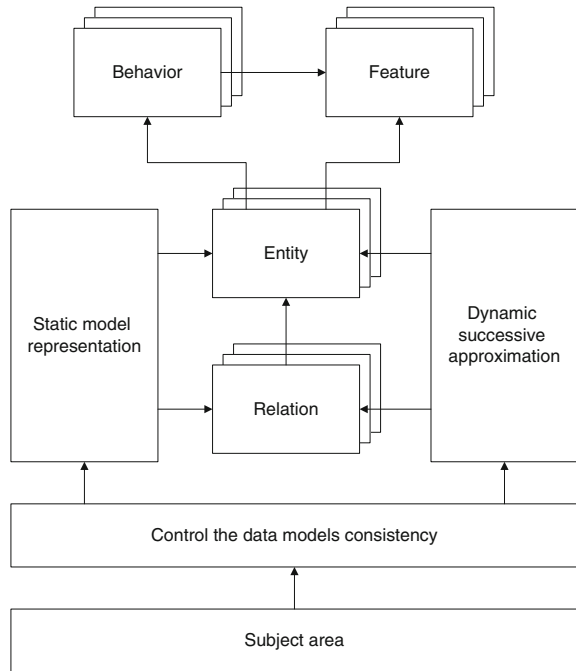
The development of modern Data Mining technologies makes claims on data repositories, which popular relational models and systems cannot respond. For instance, one of recent claims is multiple reorganization of data model during information requests [5].

The structure of obtained problem-oriented knowledge is reflected in data model used in the chosen physical storage. Usually the model reflects structures of different application spheres or subject areas and is a fundamental model for describing the suggested knowledge representation method. In general, the data model is determined by the complex of elements descriptions (entities), connection (relations) and the aggregate of these elements properties and behavior descriptions. The term "data structure" is commonly connected to its external representation. It is very important to use data structure that corresponds to the object structure for optimization of input data transformation and for its further processing effectively [6].

The suggested knowledge representation method involves static model representation and dynamic successive approximation to the user's aim. An important difference of suggested method is built-in tools of controlling the data models consistency, constructed within it during the storing and processing (Fig. 1).

The subject area has a great importance for knowledge structuring. In real world an object or a group of objects can be represented by other dissimilar objects. Usually such objects combine into a single hierarchical structure. Complex objects are implemented as constructions with simplified objects. Thus, the concept of hierarchy is essential for representation of structural relations in objects. Often hierarchy is a multilayer structure, where structural elements of the object are

Fig. 1 Knowledge representation based on object approach



constructions composed of simpler entities. In suggested knowledge representation method data hierarchical structure reflects abstract structural relations that correspond to objects being in hierarchical relations in the real world. So, a systematics is assigned in data structure, which determines the semantics, and knowledge processing software must have capabilities for these semantic relations control [7].

So, apparently, new means and methodologies for knowledge database structuring need to have objective orientation related to object selection. To create applications, reflecting subject area objects, close to real ones, it is need to research in post-relational direction.

The studied “object” philosophy is mostly popular now, because the term “object” in any object-oriented system means the combination of data and software, which represent some entity (description) of the real world. Each “object” data are composed of components of random type called attributes. Each program that describes the “object” behavior and changes its attributes is called a method. Users can not always see what the object has inside, but they can use it addressing to its program component which is assigned by abstract interface.

Recent studies results show that the most popular relational technology is not suitable for the work with complex objects. Let consider the example of the work with a complex object. This example is described in different literature sources in terms of object-oriented programming: “Hardly anyone comes with an idea to disassemble a car (complex object) into parts (entities) before placing it in the garage (database), rearrange them and collect it backwards the next morning”.

Object-oriented approach as a knowledge representation method gives an opportunity to work with complex structured data and to effectively overcome constraints connected to relational DBMS technologies. In relational databases hierarchical data is represented as many-to-many relation tuples. During extraction of data with attributes the relational database must perform “expensive” connecting operations i.e. to spend computation resources. Hierarchical data in object-oriented knowledge bases are represented, stored and processed by natural way due to the fact that attribute value can be an object.

3 Principles of Problem-Oriented Knowledge Extraction and Structuring for Object-Oriented Knowledge Database

As previously noted, the development of object-oriented knowledge database controlling systems is caused by increasing amount of stored and processed information and with the need to satisfy requirements of applications, that differ from data processing applications, which are typical for relational database systems. Attempts to use relational database technologies in such complex applications as computer aided design (CAD) systems; computer aided manufacturing (CAM); programming technologies; knowledge-based systems and multimedia systems have met a set of constraints of relational database systems. Under conditions of appearance of new database applications, new demands are occurred, which are satisfied with the use of object-oriented databases (OODB) [7–9].

The reason of object-oriented database system appearance was a need for more suitable real world entities representation and modeling, because OODB provides much more advanced data model, than traditional relational databases. The paradigm of OODB is based on fundamental concepts, like objects, identifiability, classes, inheritance, overloading and late binding [7, 10].

In object-oriented data model each real world entity is represented by one term—an object. The object has condition and behavior. Object condition is determined by its properties values—attributes. Properties values may be primitive (such as strings or integers) and non-primitive. Non-primitive ones involve a set of properties. Thus, objects may be defined in terms of other objects recursively. Object behavior is determined by methods that control object condition. Each object has a unique identifier which is defined by the system. Objects that have the same properties and behavior are grouped in classes. The object can be an instance of a single class or several classes.

The majority of OODB implementations are based on following knowledge extraction and structuring principles (Fig. 2).

Abstraction. Each real “thing” in database is a class member. A class is defined as an aggregate of properties, methods, public and private data structures and

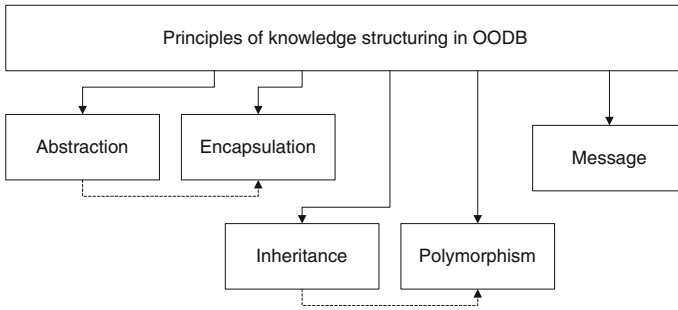


Fig. 2 Principles of knowledge structuring in OODB

managing program applied to objects (instances) of a given class. Classes are essentially abstract data types. Methods are procedures called to produce actions with the object (for instance, to print itself or to copy itself). Properties are data values related to each class object characterizing it in some way (for example, color or age). Properties do not occur in each class realizations, they are just short entries without parameters.

Encapsulation. Internal representation of data and details of public and private methods (functions or programs) is a part of a class definition and is known only inside this class. Access to class objects is permitted through properties or methods of this class or its “parents”, rather than by the use of internal realization specialities knowledge.

Inheritance (may be single or multiple). Each class is defined as a part of class hierarchy. The definition of a lower-level class inherits properties and methods of its “parent”, unless they are declared non-heritable fairly or was modified by a new definition. In the context of single inheritance a class can have only one parent class, i.e. class hierarchy has tree-like structure. In the context of multiple inheritance a class can derive from multiple parents, i.e. class hierarchy has a structure of direct non-cyclic graph, unnecessarily tree-like. Not all modern object-oriented DBMS supports multiple inheritance.

Polymorphism. Several classes can have the same names of methods and properties, even if they are considered as different. This allows us to write access methods which will properly work with objects of different classes, if only corresponding methods and properties are identically defined in these classes. For example, the “Print” method can be defined in many classes, but do different work depending on object class it is applied to.

Message. The interaction between objects is realized through the message sending and receiving responds. This is different from procedure calling for other models. To apply a method to an object we need to send a following message: “apply this method”. The paradigm of the message sending is not always used in object-oriented databases due to its complexity.

4 The Software Architecture

As a result of conducted studies we developed the software for problem-oriented knowledge search and processing on the basis of a new approach of problem-oriented knowledge representation and principles of object-oriented data storages building. Information and logical model of software environment shows the subject area data as an aggregate of information objects and relations between them (Fig. 3). The input data in top level contains mobile visualization device, physical data storage, subject area, requirement to knowledge representation results.

Creation of informational materials, 3D models and benchmarks is performed by following users' categories: programming engineer, subject area expert and designer-visualizer. Form technical functioning point of view, components needed for system work are: mobile operation systems for the application launch; free 3D models editor called Blender and benchmarks needed for control block creation. The output information for the software is results of knowledge search and processing—new problem-oriented knowledge.

The software environment architecture (Fig. 4) includes following modules: Access Management, User Management, Back-Up Master, Mark Recognition, 3D Models, Tasks Generator, Mark Calculation and Control Block. Administrator, expert and user interact with informational tool environment through corresponding graphic interfaces.

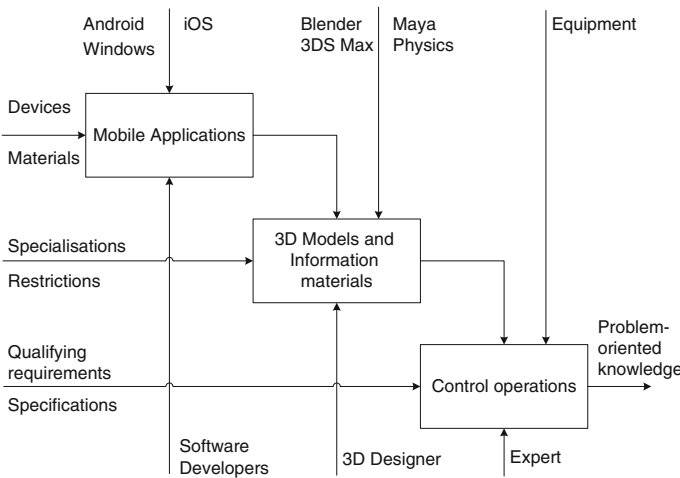
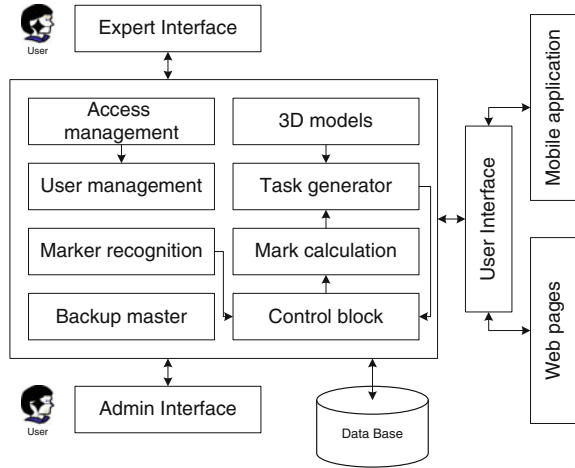


Fig. 3 Figure 3 Decomposition of information and logical model

Fig. 4 Software architecture



5 The Developed Software

The main criterion to choose a program feature implementation is a cross-platform support. This criterion exactly represents general tendencies in the development of applicable problem-oriented software. The authors chose Unity design environment. It is a design tool for 2D and 3D applications and games running Windows and OS X. Applications created using Unity can be run Windows, OS X, Windows Phone, Android, Apple iOS, Linux and Wii, PlayStation 3, PlayStation 4, Xbox 360, Xbox One. This program package allows to design applications for running in browsers with the use of special Unity plug-in (Unity Web Player) and the WebGL technology implementation. Earlier there was experimental project realization support in terms of Adobe Flash module, but later the Unity design team refused it.

Applications designed with the use of Unity support the DirectX and the OpenGL technologies that are used for the visualization of 3D interactive scenes and models. This software is used by major developers (Blizzard, EA, Ubisoft), as well as by individual developers (small project team) due to its free version, user-friendly interface and the ease of use.

The Unity has a simple Drag&Drop interface (Fig. 5), which is easy to adjust, composes of various windows that allows debugging applications in the editor. The software supports three script languages: C#, JavaScript (modified), Boo (Python dialect). The editor supports DirectX 11 and HDR. Physics calculations are performed by physics engine PhysX by NVIDIA.

The example of the software implementation is a mobile applications complex and a cloud server platform (knowledge and data storage), which allows to solve local knowledge management problems in education field (education and academic performance rating). The authorized user is asked to choose a section (subsection) and a theme (information unit) for knowledge search. If expert is not going to add a

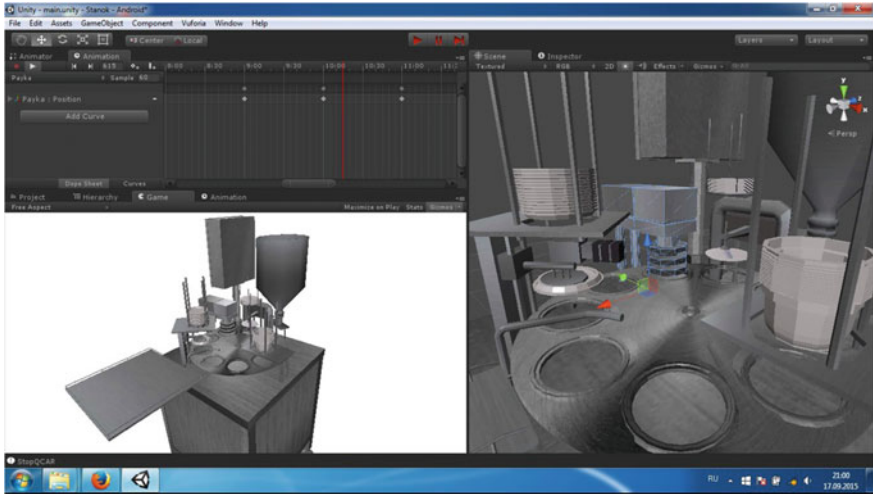


Fig. 5 Unity interface

new data of subject area (education content, 3D models, tests), the chosen section can be opened in rendering view. While viewing, the user may edit the embedded content, and then the refreshed content is accessible for review. Then the expert can shut down the application or open other sections for search and visualization.

The developed software is implemented for mobile devices. It is intended for practical lessons based on mobile applications in the education environment. The environment contains an augmented reality module permitting to increase the visualization level of studied objects and materials. The developed solution supports different user categories: the expert, the consumer (client) and the administrator. The environment gives an opportunity to improve the quality of problem-oriented knowledge representation in the current teaching content, to organize knowledge search and processing at a time suitable for the user through a single cloud storage and local client mobile interfaces. The software functioning with mobile devices excludes user reference of time and place of problem-oriented knowledge search and processing.

6 Conclusion

Considered principles of problem-oriented knowledge extraction and structuring are fundamental for computer-aided engineering of knowledge databases. On the basis of these principles we can develop a distributed information system that has a set of benefits. For example, it is the maximum proximity to the subject area data structure, that allows to increase the efficiency of information search and extracting requests executing for its further processing.

The benefits of the suggested object-oriented approach for problem-oriented knowledge representation are apparent, but relational technology has important advantages—theoretical validity and reinforcements by standards. The majority of object-oriented databases have a lack of requests executing aids. Object-oriented knowledge databases permit now to create flexible database schemes and to fill it with objects, but they cannot give us powerful tools for extracting objects from database for shared access with other users.

The developed software environment for problem-oriented knowledge search and processing in the form of mobile application complex and cloud server platform gives an opportunity to solve the local knowledge management problems effectively. The environment has shown its high reliability and efficiency while designing informational subsystems for education processes support.

In future we intend to develop new principles and approaches for creation of distributed information systems of access to problem-oriented data and knowledge and object-oriented represented knowledge.

Acknowledgments The study was performed by the grant from the Russian Science Foundation (project # 14-11-00242) in the Southern Federal University.

References

1. Gladkov, L.A., Gladkova, N.V., Legebokov, A.A.: Organization of knowledge management based on hybrid intelligent methods. In: *Advances in Intelligent Systems and Computing*, vol. 349, pp. 107–112 (2015)
2. Wu, G., Pedrycz, W., Ma, M., Li, H., Liu, J.: A particle swarm optimization variant with an inner variable learning strategy. In: *The Scientific World Journal*, vol. 2014, Article number 713490 (2014)
3. Dukkardt, A.N., Lezhebokov, A.A., Zaporozhets, D.: Informational system to support the design process of complex equipment based on the mechanism of manipulation and management for three-dimensional objects models. In: *Advances in Intelligent Systems and Computing*, vol. 347, pp. 59–66 (2015)
4. Kureichik, V.V., Kravchenko, Y.A., Bova, V.V.: Decision Support Systems for Knowledge Management, Software Engineering. In: *Intelligent Systems Advances in Intelligent Systems and Computing*. vol. 349, pp. 123–130 (2015)
5. Kravchenko, Y.A., Kureichik, V.V., Zaporozhets, D.Y., Zaruba, D.V.: Information and knowledge integration based on simulation modeling. In: *9th IEEE International Conference on Application of Information and Communication Technologies, AICT 2015*. pp. 22–25
6. Popov, L.M., Ustin, P.N.: Formation of psychological readiness for higher educational institutions student professional activity. *Soc Sci (Pakistan)* **10**(4), 418–420 (2015)
7. Gao, Q., Wang, D., Gao, F.: Impact of knowledge representations on problem-oriented learning in online environments. *Int J Human-Comput Interact.* **31**(12), 922–938 (2015)
8. Haitzer, T., Zdun, U.: Semi-automated architectural abstraction specifications for supporting software evolution. In: *Science of Computer Programming 90 (PART B)*, pp. 135–160 (2014)
9. Bova, V.V., Kureichik, V.V., Lezhebokov, A.A.: The integrated model of representation of problem-oriented knowledge in information systems. In: *8th IEEE International Conference on*

- Application of Information and Communication Technologies, AICT 2014—Conference Proceedings, pp. 111–115 (2014)
10. Lezhebokov, A.A., Kravchenko, Y.A., Bova, V.V.: Support system for QR-code-based educational processes. In: 8th IEEE International Conference on Application of Information and Communication Technologies, AICT 2014—Conference Proceedings, pp. 482–486 (2014)