UDC 004.652

O.M. Veres, Y.O. Veres
Lviv Polytechnic National University

APPLICATION OF THE OBJECT-ORIENTED APPROACH TO CONSTRUCTING DSS MODELS

This article describes an object-oriented approach of DSS modeling. Proposed and described a three-level architecture of conceptual DSS model. Analyzed the possibility of reusing components of various structure types. Described the principles of building a custom DSS by using reusable components.

Keywords – object-oriented approach, architecture, structure, model, decision making, Decision Support System.

У статті описано об'єктно-орієнтований підхід моделювання СППР. Запропоновано та описано трирівневу архітектури концептуальної моделі СППР. Проведено аналіз можливості багаторазового використання компонентів різних типів структур. Описано принципи побудови користувацької СППР з використання багаторазових компонент.

Ключові слова – об'єктно-орієнтований підхід, архітектура, структура, модель, прийняття рішення, система підтримки прийняття рішень.

Introduction. The general formulation of the problem

One of the current managers problems is understanding market opportunities and risks and in time decisions which improve useing of corporate resources. Decision Support Systems (DSS) - a computer system designed to improve the efficiency of decision-making, helps individuals who make a decision to use models and data to solve semi-structured and unstructured decision-making problems [1]. DSS - is a collection of intelligent information applications and tools which are used to manipulation and data analysis and giving the results of such analysis to the end user. Modern DSS allows to predict the impact of decisions on further business development.

The main problems which were found in earl DSS projects, are large consumer cost and adaptation complexity to rapid changes in the decision of organization support requirements. Rapid progress in information technology contributed to the research degree of universality as a basis for flexible and rapid development of DSS.

Analysis of recent research and publications

Since the late 1970s, the major progress trends are DSS generators and generalized model management system (MMS). In both cases, the basis for the versatility of DSS are functionally-oriented models.

Easy and fast growth of computer systems is not the only problem of DSS area. For example, it does not differ from the basic tasks of software development. In this field, reuse is considered as a key to improving the quality and productivity of software development, with the main accent on the potential contribution of object-oriented (OO) approach [2].

Examining existing conceptual DSS models, there are approaches based on usage the ideology of information systems, artificial intelligence, and an instrumental approach [1].

Perspective today is usage of the object-oriented paradigm to build a conceptual DSS model. Possibility of reuse is considered the key for achieving productivity and software quality. DSS - a systems designed to improve decision-making, but information technology can have a significant impact on decision making in the presence of methods that allow you to quickly and easily develop DSS. DSS

development process can be made more effective by using domain-specific reusable components that positively impact on quality of this DSS. To achieve these goals special is the contribution of object-oriented (OO) paradigm.

Structures - a complex reusable units, consisting of a number of classes that are specifically designed for improving and using as a group. The structure [3] can be considered as high-level application or architecture of subsystem that consists of a set of classes that are specifically designed to be improved and used as a group. The structure is a total solution for developing applications in a particular area (eg, image editors, operating systems) that can be customized according to the problem. Domain-specific components, in which structures are used to represent classes of decision problems like prototype of decision-making elements, is a basis for solving problems of DSS development and design [3]. This approach, like a simulation tool, designed for decision maker (the person who makes decisions). The structure is considered as a general model of decision making for a particular class of problems that can be configured through the process of implementation to provide features available decision problem. Specific models can be built by selection, adaptation and problem domain concepts combination.

Unsolved before aspects of the main problem. The model-oriented DSS as a tool to support decision making managers are limited. Model development as usually carried out by specialists, and managers often feel reluctance to use models that they do not fully understand, and in development which they did not participate. It is reasonable to develop simulation environment for users who are not simulation experts using reusable components.

Article objectives (tasks)

The main objective of the paper is development DSS conceptual model and its components guided by the object-oriented paradigm. The approach is designed for professionals (ie software developers with OO mechanism knowledge) that have to create a DSS through structures reuse.

Main material

No general solution can be built without an understanding of the nature and characteristics of the specific actions that it generates. Agreed functional architecture of DSS is a structure with three major functional components, namely model component (MDS), (DBMS) and dialogue. DSS such functional architecture is called model-oriented DSS.

Object-oriented approach used by professionals (software developers, specialists with knowledge of OO mechanisms) for DSS creation through reusing structures. One of the main objectives in DSS investigation is to provide a simulation environment for a user who is not modeling expert [4]. The basic idea of using reusable components is that using a component-oriented application design DSS development model for decision maker becomes a simple process of selecting and applying specific complex, domain-specific concepts to describe the solution of the problem situation [3]. It is possible to distinguish the features of a custom DSS (user DSS), namely the modeling paradigm; visual representation; recommendations.

Decision maker has to submit his specific problems at the conceptual level. This is possible if the object-oriented building blocks used to construct DSS offered to decision maker as the *modeling concepts*. Model construction is to set up a general decision making model for decision making problems classes by implementing concepts and linking default blocks. For example, IT project budget investment class. Project life may continue for several years and foresee prediction of all income and expenses that are made in the period. Return on investment can be measured by some criteria, such as: the net present value (NPV), internal rate of return, payback period, etc. Models for this decision problems class can be structured and formulated in terms of concepts such as IT project life, the schedule of it revenues and expenses, cash flow, profitability criteria, cost of capital and so on.

For a visual representation of concepts, are using structure representation as tables, graphs, reports, charts and so on. Using those visual images, users can create and manage model concepts examples.

Recommendations for the model development are given by dialogue component. They should help in concepts selection for model development and implementation, and combinations of units. Recommendations are also required for model performing.

Given the objectives of DSS and features of the object-oriented paradigm for the design, conceptual DSS model can be represented in three levels architecture: the level of semantics, presentations and recommendations (Fig. 1). This architecture fits interactive applications requirements, which promote independence between application semantics and interface [5]. The difference between DSS architecture levels allows identifying the various duties which have sets of classes that represent each part as follows:

- **semantics level**: its classes are used to describe corporate solution directed to solving problem class. Proposed decision making problems modeling methods and their solutions by questions and decision procedures;
- **presentation level**: its classes are used to describe the structure of overall presentation (tables, graphs, etc.); can be used for visual presentation concepts presented at semantics level. Proposed methods for solving conceptual value presentation structure, and its graphical representation;
- **recommendation level**: its classes cover all aspects conversational interface that defines scenarios interaction for simulation and action taking.

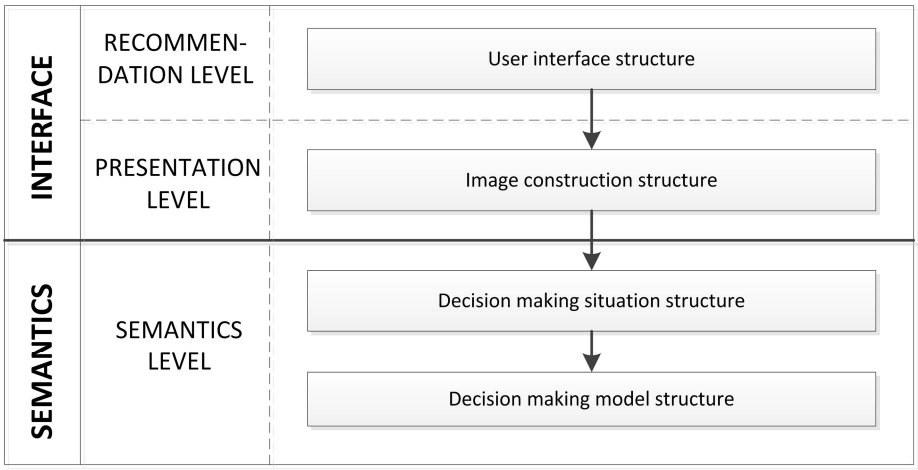

Fig. 1. Conceptual DSS model (object-oriented approach)

Architecture can be viewed as a higher-level structure for building DSS purposes, which consist of lower-level types of structures and principles of their combination. The structure acts as a "specialized methodology", facilitating the activities of development proposals, describing in detail the nature and characteristics of different units and their interconnection.

Semantics level reusable components. Solution models are an elements representation that characterizes decision making problems, for searching analyzing and evaluating possible solutions. Solution models is it is always incomplete and approximate reality representation. It represents a *special approach* to analyze the original problems of decision making, and may be represented in accordance with the successive *abstraction levels*. By analogy with the software development industry, decision making models are reviewed according to three main abstraction levels, namely the *specification* (problems area), *design and implementation* (solution area). Solutions area in DSS field related to the method selection or problem *solving* technique. Each solution technology includes *modeling paradigm* consisting of a set of specific concepts and relations, according to which must be structured problem solution. DSS problem area outlines decision making situation characteristic, regardless of its solving method. One of the ways to get closer to problem solving at the specification level is to use existing decision making solving theories. The concept and practice of existing theories form the *theory domain.*

In order to develop the DSS semantics level identified two structure types, namely the decision making situation structure and decision making model structure.

Decision making situation structure provides a common model development paradigm, which focuses on a special class of problems. The structure of the decision making situation - a set of related classes that represent subject-oriented modeling target concepts derived from the theory domain. Universality degree of decision making situation structure regarding on problem solution class depends on the *invariants* that can be defined in model problem solution of this class with concepts and relations that belong to the theory domain.

In representation of these modeling invariants, structure elements classes of decision making situation, is an abstraction, or:

- borrowed concepts and relations that belong to the theory domain;
- factor from set of concepts general characteristics to represent a higher level of abstractions;
- added specially for targets structuring or to provide greater flexibility in models construction.

Basic methods allow you to create, update and delete concepts (samples, examples), and receive information associated with them. Common questions are determined by methods to outline problems in this class. For example, typical task to resolve the formation of IT project investment budget is "NP computing", "increase the Net Present Value (NPV)" and so on. Most methods determine the questions to be abstract, since the decision making situation structures are independent from decision making model structure.

Decision making model structure contains general modeling paradigm, which lies at the basis of the decision making method, and algorithmic details of this method. In the simulation paradigm, a set of classes that construct the solution model structure, which reflects the basic concepts, relations and assumptions required to develop a decision-making model in decisions space. In addition, the modeling paradigm also contains methods that either implement algorithms to perform sensitivity analysis and models optimization [5, 6], or interface to perform the available DSS generators/applications functions. The second option is based on the fact that on the market already available and widely used resolution methods tools (spreadsheets, MS/OR, statistical packages, etc.). Structures may also contain additional classes for representing data structures required by the application solving algorithms and conversion procedure [6-8]. A simplified example of the decision making model structure is descriptive algebraic model (DAM). It is based on directed causal relationships among variables specification. OAM provides solution methods, where the values of variables or accurately represented, or is deduced by using relations. OAM can be watched as a set of variables, where each variable is characterized by type, dimension and variable operations set.

Presentation level components reusing. The main purpose of the *presentation* level to get concepts from level semantic that are important for the description and execution of this class of problems, providing DSS users with consistent and uniform models appearance that are compatible with their cognitive world. This work is based on the premise that the current known instrumentality base for presentation structure representation which are known by decision maker and can be simply adapted and used as a visual representation of different problem solutions or decision maker most common presentation structures are tables, forms, hierarchies, graphs, algebraic formulas and more. Basic structure describes a number of concepts, relations and behavior, regardless of their specific usage. They define a background of the structures main visual properties and manipulation opportunities. Static and dynamic properties of structure presentation classes does not affect on the semantics.

Recommendation level components reuse. *Recommendations* level covers all conversational aspects of "man-machine" interface which define the scenarios interaction for model implementation and usage. Interface "man-machine" field is one of the most prolific for the reusable components development. Can be used as elements of OOPLs (windows, dialog boxes, buttons) low-level class libraries, and larger user interface elements.

Any of them are suitable for the recommendation level, given that it requires reusable components for dialogue development.

Structures composing for DSS development. Investigated decomposition process which allows to isolate structures at different abstraction levels and with different roles based on a clear definition of target DSS.

Applications development is the compiling process, where reusable components will be improved and combined, to develop specific DSS, according to a multilevel architecture shown above.

We use the term assembly (composing) just to emphasize the fact that in DSS context those classes belong to different structures types [9, 10]. At the semantics level composing process may consist of three aspects: assembling various decision making situations structures, selection different decision making structures, decision making situation structures(s) assembly within the decision making model structure(s).

The first kind collection purpose is expanding the boundaries of the chosen subject area through the subtask integration. For example, a DSS investment budget can be viewed as a set of next related tasks: separate investment projects analysis, alternative investment projects with limited capital choosing, estimation of capital cost, funding sources choosing. Each task is represented as a structure. Reusable components usage is also possible.

The purpose of assembling the decision making model structure is to provide as many solution paradigms to manage with the needs of decision-making structure according to solving questions. The third assembly type due to the fact that the decision making situation structure should always be collected, at least the decision making model structure, in order to ensure that the formulated models can be analyzed (ie, performed). The goal is to establish a correspondence between solutions concepts and their "equivalent" representation according to the decision making, as well as to automate the switching from one to another solution. The basic idea is that the creation of solution concept instance initiates the appropriate instance on the communication level, which makes it possible to establish a behavior chain and communication at every architecture level. Similarly, other operations that characterize the decision making objects behavior modeling will cause the appropriate objects behavior modeling. Farther, questions formulated at the highest level must be associated with the methods that are actually caused by running procedures at the solution model level (solution methods).

Decision-making models have visual representation and because of this visual representation, users can manipulate models (design and run). In accordance with the tasks class characteristics and cognitive investigations, one or more presentation structures are selected as components of its visual representation. Image constructing structure will be constructed from DSS semantics structures parts to link general purpose components of image structure. Users thus create and manipulate modeling concepts instances by visual representation they are given. Thus, each operation is used to represent objects, solves the problem of its representation and corresponding to decision making object.

Dialog for developing and executing scripts is developing with basic use of reusable user interface components. Clearly, any new classes that are required to implement additional features (semantics or DSS interface), which is not covered by structures, can be added optionally.

Conclusions

To achieve this goal conducted the analysis of the application of object-oriented approach to quickl and easy DSS designing which is based on various structures types reusing. Proposed a three-tier DSS conceptual model architecture which reflects reusable components evolution, which have a different nature, namely decision making situation structure, decision making model structure, image construction structure and user interface development structure. Describe reusable components types and sequence custom DSS composing. This approach can be considered as higher level structures designing, which consist of lower level structures and recommendations with these structures usage to develop specific DSS.

Further research will focus on methods research, models and tools to more effectively support the overall model development, namely the knowledge extraction methods from separate areas and structure storages.

1. Верес О.М. Компоненти концептуальної моделі системи підтримки прийняття рішень / О. М. Верес // Комп'ютерні науки та інформаційні технології. Вісник НУ "Львівська політехніка".

– 2010. № 686. – С.103-112. 2. JOHNSON, P. Object-oriented technology: the competitive advantage. GEC, 9(1): 28-41, 1993. 3. NIERSTRASZ O. Component-oriented software development / O. NIERSTRASZ, S. GIBBS, D. TSICHRITZIS // Communications of ACM, 35(9): 160-165, Sept. 1992. 4. BECKER K. Reusable object-oriented specifications for decision support systems / K. BECKER, F. BODART // In: IFIP WG 8.1 Working Conference on the Object Oriented Approach in Information Systems - Quebec City, Oct. 28-31, 1991. Proceedings. North-Holland, 1991. p. 137-155. 5. HARTSON H. Human-computer interface development : concepts and systems for its management / H. HARTSON, D. HIX // ACM Computing Surveys, 21(1):5-93, March 1989. 6. DOLK, D.R. & KONSYNSKI, B. Knowledge representation for model management systems. IEEE TSE, 10(6): 619-628, Nov. 1984. 7. BLANNING, R.W. A relational theory of model management. In: HOLSAPPLE, C.W. & WHINSTON, A.B. (eds.). Data base management: theory and applications. Springer- Verlag, 1987. p. 15-53. 8. MA, J. An object-oriented framework for model management. Decision Support Systems, 13(2): 133-149, Feb. 1995. 9. MUHANA, W.A. An object-oriented framework for model management and DSS development. Decision Support Systems, 9(2): 217-229, Feb. 1993. 10. MUHANA, W.A. SYMMS: a model management systems that supports model reuse, sharing and integration. European Journal of Operations Research, 72(2): 214-242, Jan. 1995.