Volume 1, Issue 4, October 2014 ISSN: 2348-8565 (Online)
International Journal of Modern Engineering and Research Technology
Website: http://www.ijmert.org Email: editor.ijmert@gmail.com

Object Oriented Framework for MCDM and DSS Modelling

Dr. Mamta Lambert
Associate Professor
Department of Computer Applications
Jabalpur Engineering College Jabalpur,
Jabalpur (M.P.) [INDIA]
Email :- lambert_mamta@yahoo.co.in

ABSTRACT

The undertaking of Multicriteria Decision Making (MCDM) and the development of Decision Support Systems (DSSs) tend to be complex and inefficient, leading to low productivity in decision analysis and DSSs. This study has developed a framework based on object orientation for MCDM and DSSs modeling. The framework can create models to represent decision problems and decision making procedures in a simple and transparent way. The practical application of the object-oriented approach in the development of DSSs and in the facilitation of decision analysis for MCDM problems has shown that the object-oriented approach can provide a solid methodological and philosophical bases for both decision making and the development of DSSs.

Keywords:— *Object orientation, MCDM, DSSs Modeling.*

I. INTRODUCTION

Decision analysis and Multicriteria Decision Making (MCDM) is the study of decision making for problems with multiple objectives, has been developed and widely applied in solving complicated decision problems. Over the years, many approaches and underlying theories have been developed for solving decision problems with multiple criteria. Over the past three decades, MCDM has experienced a striking development period both in practice and theory and has developed into a discipline in its own right. Since the 1980s, one of the trends has been a shift towards the implementation of computerised Decision Support Systems (DSSs). A Decision Support Systems (DSS) was first characterized as "a computer- based information system" to facilitate decision making activities. Some multiple criteria DSSs have been implemented to improve decision making in originations. Solving decision problems via decision analysis can be divided into several steps : problem analysis, problem structuring, evaluation, choice and implementation.

In MCDM and DSS development, the issue of philosophy and methodology is most important. Keeney points out that a philosophical approach and methodological help is missing in most decision making methodologies to understand and articulate values and to use them to identify decision opportunities and to create alternatives. Another important issue is to find effective and efficient methods for decision analysis and DSS development. In this paper, these two issues are dealt with by the methodology proposed, based on object orientation. The methodology provides a uniform framework for decision analysis and DSS modeling. The framework can create models to represent decision problems and decision making procedures in a simple and transparent way,

reuse existing knowledge and thus handle the complexity and inefficiency in both decision analysis and DSS development.

Object orientation has been involved in some aspects of DSS implementation, such as programming, architectural design, model management and user interface development. In this paper, the methodology and the philosophy based on object orientation for decision analysis, especially in problem analysis and structuring, and DSS development are discussed.

II. Object Orientation

The methodology of object orientation originated from object-oriented programming, which in turn originated in simulation modeling aided by the development of the discrete event simulation language, Simula, in the late 1960s. Interest shifted to object-oriented design after object-oriented programming began to mature. Object-oriented analysis became a major area of attention in the late 1980s.

Basic Concepts of Object Orientation

The common element in object orientation is the "object". In object-oriented systems, all real world entities, either concrete or abstract, are treated as objects. The most general version of the object paradigm defines an object as a logic machine of whatever level of granularity, which may be interconnected with other logic machines to construct a system. The objects in a system may intercommunicate by receiving and sending messages to each other. Objects in the world are categorized as a hierarchy. A set of object instances of the same nature is collectively organized as a "class". Objects can be represented as.

Class (name of the class, for example, Human being class)

- instance (name) (the particular occurrence of the class, for example, "Jones")
- services (operations) (for example, "Sleep")
- arguments (attributes) (for example, sex, date of birth)

Objects encapsulate different properties of the entities, such as arguments (attributes) and services (operations or behaviours). The attributes and operations encapsulated in an object can only be accessed by passing messages to that object. Encapsulation is one of the fundamental ideas behind the object-oriented approach.

Object orientation views the world as made up of objects and messages. An object is a complete entity which performs a definite task and contains all components needed to carry out the task. The link between an object and its environment, other objects, is made via messages. The effect of a message on an object depends both on the message and the receiving object, that is, the same message may cause different actions from two different objects. The actions of individual entities and those of the community or of the world as a whole are invoked by the "messages" among the objects. Objects communicates by passing messages. Objects respond to messages by selecting a corresponding method to execute the message received. "Method" are also called "operations" or "behaviours" in the objects.

Besides encapsulation, another major feature of object orientation is "inheritance". "Subclasses" can inherit the properties of their "superclasses" and can also have some special properties of their own. Child classes may inherit some features from the parent classes while being able to override other attributes. Inheritance of classes may simplify the system by reducing the number of independent system components that need to be identified. It also supports the interactive process of step-wise refinement with encapsulation and information

hiding by allowing us to defer the definition of internal detailed activities of a class until necessary.

Advantages of Object Orientation

Reusability – Reusability indicates that the system classes can be reused for the implementation of similar systems.

Extensibility – Extensibility Means that a system can be easily extended by adding classes as basic building components of the system.

Because of these advantages of object orientations, it seems that there are many benefits obtained from the application of object orientation in modeling decision making and DSSs.

As asserted by Graham, the benefits at the analysis and modeling level of object orientation are potentially the greatest among analysis (modeling), design and programming. Analysis specifications may be at least as reusable and extensible as object-oriented programs since the former is often less specific to a purpose.

III. Methodological Background

Object orientation may provide a philosophy and a methodology which cater for both decision analysis and DSS modeling. Object orientation offers a philosophy to naturally model the real world. An object-oriented philosophy holds the view that the world and its constituent parts are composed of independent yet interactive physical or non-physical objects. Decision problems and DSSs can naturally be modeled by a collection of objects. Object orientation permits a system to be described in the concepts of the real world. Partitioning a problem domain into objects corresponding to a concept-oriented view of the real world counterparts is often more natural than a functional decomposition. Object orientation may contribute to the productivity of decision analysis and DSS development in two ways. First, Object orientation will be able to utilize the research outcome from the literature and the experiences from the previous case studies by the mechanism of reuse. Object orientation offers a formal mechanism, which is the inheritance of reusable objects, to reuse the existing proven knowledge and past experiences of a similar decision context. Secondly, an object orientation provides a uniform tool to deal with almost all the aspects of decision making and DSS development. Object orientation will allow these phases to be carried out in a uniform and coherent way.

Object orientation also allows the accumulation and reuse of knowledge and experiences in various aspects of decision analysis. In object orientation, objects with the same or similar properties are abstracted as a "class", which is a kind of "taxonomy" and is reusable in specific cases of decision analysis. Problems, problem elements and DSS components are classified as classes. An object is an instance of a certain class and would automatically inherit the general characteristics of the class. Reusable classes store knowledge and experiences of decision analysis and are very important for a specific decision problem to be structured for a DSS to be developed.

IV. Object Oriented Problem Structuring and Analysis

Problem structuring and analysis are the initial activities in any decision analysis, whose success is largely dependent on the right identification and definition of the decision problem. Problem structuring also known as problem modeling is an imaginative and creative decision making process of translating an initially ill-defined problem into a set of well-defined elements, relations and operations. Bana e Costa, Stewart and Vansnick suggest a detailed definition, which they believe might find wide consensus. According to them, the structuring and framing

of a decision situation is a constructive and learning process.

Figure. 1 shows the overall diagram and the general process for the methodology of object-oriented problem analysis and structuring. "Context" stands for the decision context in which the specific problem needs to be solved in a MCDM way. Context classes represent a category of decision contexts with similar features or in the same field of research such as a specific kind of natural resource management. The instances of the context classes are context objects, which indicate individual problem circumstances. People involved in decision analysis, such as decision makers. People analysis is mainly focused on the roles played by the people involved in the decision problem and decision making activities. Decision elements are entities that are included in the decision problem under consideration. Criteria, alternatives and decision makers are examples of decision elements. Decision element classes represent categories of decision elements with similar attributes. The identification of various decision elements provides the basis for problem analysis. Decision criteria, for example are structured as hierarchies. Decision alternatives are generated out of the action elements defined. The communication mechanism is determined among various decision participants. The Methodology is human oriented. People are analysed and modeled at the initial stage of problem analysis and structuring.

The methodology utilizes reuse, one of the major benefits from object-orientation, in decision analysis contexts. This is done through the instantiation of classes. Classes may represent the existing knowledge and experience of decision contexts and decision elements while objects are related to the specific problem situations. Reuse is how ever not the necessary basis for our method as the methodology can be used in the very first problem cases, in which there are no existing classes.

V. Object-Oriented DSS Development

A reuse repository is a central resource for the whole development process and can be searched for reusable components at each stage. An easily accessible repository is essential to support the reuse process through analysis, design and implementation of DSSs. A two-level reuse repository is suggested by Gossain. One level contains verified components that have undergone qualification and selection criteria to ensure that they are adequately tested and documented. The other level contains those components that have not been verified. These include not only reusable codes but also the storage medium for previous business models, process models, analysis models, design models and other documentation from past projects. It is a place for shared knowledge and is a kind of classification of the system development knowledge. A four-level framework that consists of eight parts for the reuse repository is proposed as shown in fig. 2. It offers a guideline for the development procedure of DSSs.

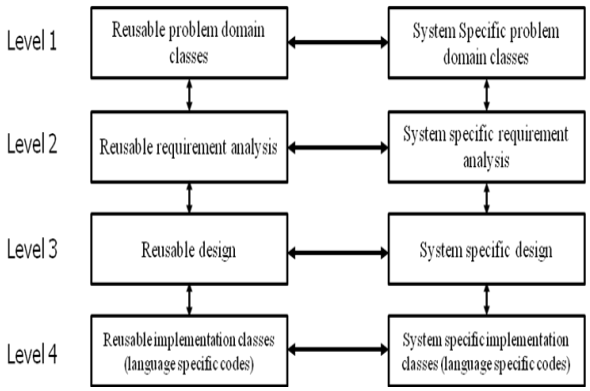

Figure 1. Object-oriented problem structuring and analysis

VI. Conclusion

The present paper has discussed the methodology and the philosophy of object-oriented decision analysis and DSS

developments. First of all, object orientation can model the real world naturally. Object-oriented decision analysis and DSS modeling offer an easy way to analyse decision problems and develop DSSs. Classes (objects) are the media for the representation of the concepts in the world. Secondly, object-oriented decision analysis and DSS development provide a uniform methodology for decision analysis and DSS development and can produce a better understanding of problem analysis and DSS development activities. Object-oriented analysis and DSS development are able to improve the efficiency of decision analysis, especially problem analysis and structuring and DSS development. There is a need to point out that this paper only introduced the main points of the philosophy and the methodology of the object-oriented decision analysis and DSS development. The frameworks, shown in the diagrams of object-oriented problem analysis (structuring) and DSS development.

REFERENCES:

[1] D.A. Adams, J.F. Courtney Jr., G.M. Kasper, A process oriented method for the evaluation of decision support system generators, Information and Management 19 (4) (1990 Nov.) 213–225.

[2] C.A. Bana e Costa, T.J. Stewart, J.C. Vansnick, Multicriteria decision analysis: some thoughts based on the tutorial and discussion sessions of the ESIGMA meetings, European Journal of Operational Research 99 (1) (1997 May) 213–225.

[3] V. Belton, T.J. Stewart, Multiple Criteria Decision Analysis: An Integrated Approach, Kluwer Academic Publishing, Boston, 2002.

[4] V. Belton, S.P. Vickers, VISA: Visual Interactive Sensitivity Analysis for Multiple Criteria Decision Aid. User Manual, Publ. V. Belton and SPV Software Products, UK, 1989.

[5] H.K. Bhargava, S. Sridhar, C. Herrick, Beyond spreadsheets: tools for building decision support systems, Computer 32 (3) (1999 Mar.) 31–39.

[6] G. Booch, Object-oriented Design with Applications, Benjamin Cummings, Redwood City, CA, 1991.

[7] D. Buede, Superior design features of decision analytic software, Computers and Operations Research 19 (1) (1992 Jan.) 43–57.

[8] J.S. Dyer, P.C. Fishburn, R.E. Steuer, J. Wallenius, Multiple criteria decision making, multiattribute utility theory: the next ten years, Management Science 38 (5) (1992 May) 645–654.

[9] T. Gal, T.J. Stewart, T. Hanne (Eds.), Multicrteria Decision Making: Advances in MCDM Models, Algorithms, Theory, and Applications, Kluwer Academic Publishing, Norwell, MA, 1999.

[10] M. Ginzberg, E. Stohr, Decision support systems: issues and perspectives, in: M. Ginzberg, W. Reitman, E. Stihr (Eds.), Decision Support Systems, North Holland, Amsterdam,1982, pp. 9–32

[11] S. Gossain, Object Modeling and Design Strategies, Cambridge Univ. Press, Cambridge, 1998.

[12] I. Graham, Object-oriented Methods, 2nd ed., Addison Wesley, New York, 1994.

[13] M.T. Jelassi, M. Jarke, E. Stohr, Design in generalized multiple criteria decision support system, Journal of Management Information Systems 1 (4) (1985 Spring) 24–43.

[14] R.L. Keeney, Value-Focused Thinking: A Path to Creative Decision making, Harvard Univ. Press, London, 1992.

[15] H. Mintzberg, D. Raisinghani, A. Thoret, The structure of unstructured decision processes, Administrative Science Quarterly 21 (2) (1976) 246–275.

[16] B. Selic, G. Gullekson, P.T. Ward, Real -Time Object-oriented Modeling, Wiley, New York, 1994.

[17] H.A. Simon, The Shape of Automation, Harper and Row, New York, 1965.

[18] H.A. Simon, The New Science of Management Decision, 3rd ed., Harper and Row, New York, 1977.

[19] P. Sully, Modeling the World with Objects, Prentice Hall, New Jersey, 1993.

[20] D. Von Winter feldt, Structuring decision problems for decision analysis, Acta Psychological 45 (1980) 71–93.

* * * * *