Journal of Applied Business Information Systems

http://www.jabis.ro

Decision Support Systems Development: a Methodological Approach

Claudiu Brandas*

* West University of Timisoara, Faculty of Economics and Business Administration, Romania

Abstract: *The analysis and design of Decision Support Systems (DSS) is a complex process that implies the usage of adequate methodologies, methods and tools for modeling decision processes. The organizations can implement DSS through customized developing according to the specific features of decision activities or by purchasing a generalized DSS, that later would be customized. In this paper we present an analytical study to identify what are the most appropriate methodologies, methods and techniques for developing DSS.*
Keywords: *Decision Support Systems, Development Methodologies, Methods, Tools, Information Systems.*

1. INTRODUCTION

Conceiving, developing and implementing DSS have known an evolution closely linked to the possibilities of implementing decision models and requirements.

Since the moment of underlying the DSS by G. A. Gorry and M. S. Scott Morton [7] the paradigm of developing the DSS has oscillated between conceiving and implementing support systems for structured problems and the one for unstructured problems within decision processes.

The analysis and design of DSS is a complex process that implies the usage of adequate methodologies and instruments for modeling decision processes. The organizations can implement DSS through customized developing according to the specific features of decision activities or by purchasing a generalized DSS, that later would be customized.

According to Marakas, there are two strategies for developing DSS [11] [12]: (1) developing DSS specific for the organization, by programming languages; (2) developing DSS through DSS generators. In addition to these two strategies, a series of organizations purchase DSS particularized for certain activities that they adapt and customize according to the requirements of the organization.

2. RESEARCH METHODOLOGY

The research that has been done is based on an analytical study of most known and used methodologies, methods ant tools for Information Systems and Decision Support Systems development. *Research problem* of this study is to identify what are the most appropriate methodologies, methods and techniques for developing DSS. *The research is exploratory and based on documentary study*.

3. FINDINGS AND DISCUSSION

Traditional development of information systems based on *systems development life cycle (SDLC)*, was the first concept in DSS development and implementation.

System development life cycle (SDLC) can be seen as founder concept of the all after coming methodologies which were developed in the information technology field.

DSS development and implementation, implies major organizational changes. In our opinion, the development of DSS from SDLC is not the best

option because of its rigidity and the huge volume of project documentation necessary.

One alternative to the SDLC concept is the *ROMC analysis* [13]. According to this model, the system analysis focuses more on representations (R), operations (O), memory aids (M) and controls (C).

The ROMC concept used for DSS design, assumes that the system analyst will investigate and create models for the existing representations which will be used as communication tools between the DSS and the users.

The ROMC concept is very useful for designing the DSS interface.

The *Functional Category Analysis* is another DSS design method [11]. This method reveals the necessary functions for design of DSS. The functions are selected form a list of functions such as: selection, aggregation, estimation, simulation, equalization and optimization.

This method has its best utility in the knowledge-based and model-based design.

According to Marakas, the general DSS development process is composed from the following elements or phases [11]:

- DSS objectives and resources identification
- System analysis
    - Functional Requirements
    - Interface Requirements
    - Coordination Requirements
- System design
- System construction
- System implementation
- Incremental Adaptation

The *RAD* method is characterized by a rapid and iterative development of the information systems, using prototyping and CASE instruments.

Major characteristics of this method are:

- Combines the best techniques and procedures for the system's rapid development. Examples: Brainstorming, prototyping and CASE instruments.
- The design is made with the direct contribution of users, advancing the system evaluation possibility, before the system will be delivered.
- Sequential and iterative implementation of the system. The system is developed in stages, so as the user's suggestions and needs to be implemented in the right stage.

Using the fourth-generation language (4GL) for DSS development. The most used languages are: Visual Studio .NET (Visual Basic, C#, C++, J#, ASP), Java, Borland Delphi, Borland JBuilder, etc.

The current information system development is based on the RAD concept. In the '90, the XP (Extreme Programming) methodology was created by Kent Beck, which was developed during several software projects [3]. This technology stands for a rapid and successive development of information system, realized by a direct implementation in the production and by a permanent focus on users' needs and requests. Moreover the system is developed in production through short interval analysis-design-programming-testing iterations.

From our point of view the RAD or XP methods are the most recommended in the development and implementation of DSS.

Arguments for our standing point are:

- The unstructured form of the problems in the decision process creates a complex and dynamic ground for the DSS development.
- The development of system based on prototyping, combined with the XP concept allows a quick implementation which will have a low risk for the DSS system.
- DSS development through RAD and XP technology allows some 4-th generation programming languages to be used, in order to create an integrated platform for the whole organization.
- A quick development of group DSS is possible due to the iteration technique, prototypes and a permanent contact with users (decision makers).

The use of prototyping is the most common method for a rapid development of the information systems. It can be found in today's almost all of techniques and methods used to develop information systems. This method is known under the name of iterative design or evolutionary development of the information systems [15].

The use of prototyping in developing DSS has become, in the last years, a common activity in the DSS designers' community. The development of the DSS is focused on a specific, dynamic and complex activity, which requests a series of updates and testing. This may be one of the reasons why the prototypes method has become so popular in the last years.

In the literature one can find two concepts related to the DSS development through prototypes [14] [11]:

- To create the system based on a throwaway prototype.
- To create the system based on an evolutionary prototype.

Current tendency related to the complexity of the DSS, imposes an integrated and rapid design process, which has to be oriented towards a reprocess of the system's components. The answer to these challenges can be found in the Unified Process (UP) Methodology.

The UP methodology for software development was developed and published by the UML language creators in the beginning of 1999 [9]. Today this methodology is the standard for the development of information systems because it is a:

- methodology for developing complex information systems;
- fundamental framework for UML, a framework in which the users requirements are satisfied and system modeling is done through diagrams which are easy to transform into software;
- framework for the implementation of CASE (Computer Aided Systems/Software Engineering) tools used for the analysis, design and UML modeling;

The unified process is structured on two dimensions [9]:

- temporal – which splits the life cycle into phases and iterations;
- process components – which produces a specific set of results as the base for a well define set of activities;

Configuring the projects along the temporal side implies the following steps:

- *Conception* – means to create a vision about the project.
- *Elaboration* – means to plan all the activities and resources; to define the characteristics and to design the architecture.
- *Construction* –means the actual construction of the product, performing several increment iterations.
- *Transition* – means product delivery to the user.

In the unified process methodology each iteration has a work flow composed from five steps [9]:

- *Requirements* – assumes the definition of system and users requirements.
- *Analysis* – assumes prerequisites structuring, modeling and finalizing.
- *Design* – assumes that the prerequisites are translated into the system's architecture.
- *Implementation* – assumes software programming.
- *Testing* – assumes testing the functionality of the system.

In practice a commercial version of the UP exists and it is called RUP (Rational Unified Process), created by Rational and developed currently by IBM .

The proper conditions under which this methodology shall be used in developing DSS are:

- the necessity of processing a huge and heterogeneous volume of data;
- a large number of users, present at all levels in the organization;
- dynamics of decision process models;
- the use of complex hardware and communication tools;
- the used of internet and intranet as a source of data and information;
- the need for a rapid development of the system;

The unified modeling language (UML) embodies a standard for visual modeling, for structuring documentation, requirements and information or other kind of systems. The UML comprise the best unified techniques which exist currently for developing complex information systems.

UML is a modeling language and not analysis and design methodology [1]. UML offers systems designers a vast thesaurus and a set of rules for conceptual and physical modeling and communication [4].

UML is independent of the programming languages, however it is commonly used in an object based information system development. The object-oriented methodologies as such the unified process methodology are the methodologies where the UML has become a vital necessity. Furthermore, the standardized level of the UML has led to CASE tools with a high performance . The instruments are used to create an interactive system. The interactive system will constitute the base for an automatic code generators into programming languages (JAVA, Visual C++, C#.NET, VB.NET, etc.).

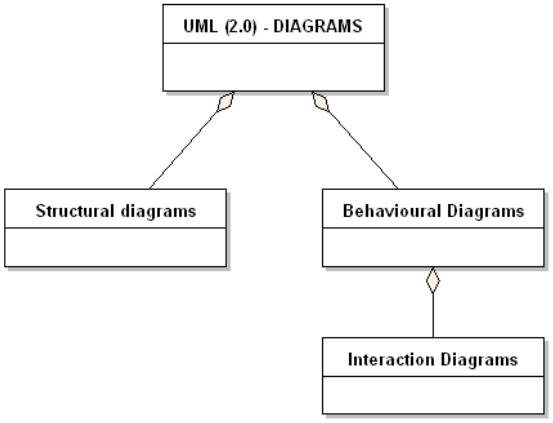

Figure 1. UML Diagrams (version 2.0).

The UML thesaurus comprises a series of diagrams and symbols which can be used to represent the system's model. UML diagrams (figure 1) are tagged in three categories :

- Structural diagrams: these diagrams show the elements of system specification which is time independent. This category comprises: class diagram, object diagram, package diagram, components diagram, composite structure diagram, deployment diagram.
- Behavioral diagrams: these diagrams show the behavioral characteristics of a system or a process. This category comprises: use case diagram, activity diagram, state machine diagram, and the four interaction diagrams.
- Interaction diagrams: these diagrams are a subset of behavioral diagrams which are used to underline the interaction between the system's objects. This category comprises: interaction overview diagram, timing diagram, communication diagrams and sequence diagram.

We conclude that by using UML in DSS development, design and implementation:

- We may better detect the decision process prerequisites.
- We may better detect the interaction between the decision factors, external environment, existing software and other elements.
- We may realize a better and faster communication between the DSS functions.
- We may create a standard for the development process and a pattern for the decision process. (decision making process for the financial accounting activities, decision making process for marketing activities).
- We may increase the speed and the rigorousness of the development process by using CASE tools.
- DSS rapid prototyping and decrease the time needed for design and implementation

The DSS development process assumes that a collaborative environment to facilitate and manage group decisions will be created and implemented.

Collaboration Engineering [5] is a new approach in the field of group decision support, which combines the design, modeling and implementing into recurrent collaboration processes, with the scope of creating a routine for some recurrent tasks, using techniques and tools to facilitate group decisions.

Collaboration Engineering is an approach that designs, models and deploys repeatable collaboration processes for recurring high-value collaborative tasks that are executed by practitioners using facilitation techniques and technology. Collaboration processes designed in Collaboration Engineering are processes that support a group effort towards a specific goal, mostly within a specific timeframe. The process is build as a sequence of facilitation interventions that create patterns of collaboration; predictable group behavior with respect to a goal [KOLF06]. Toward that end, researchers have begun to codify a collection of such building blocks, called thinkLets

A thinkLet is a named, packaged facilitation technique that creates a predictable, repeatable pattern of collaboration among people working towards a goal [6].

The thinkLet technique can be employed in the design of collaboration processes between

different domains. A set of thinkLet sequentially executed in a collaborative process, forms a decision module. A special characteristic of these modules is the unified processing of distinct sets of information (for example: the opinions and contributions of the participants for a particular subject or problem). The execution of one such module will lead to a distinct result in the context of a group decision process (for example: a priority list of options, the group agreement over a concept or a procedure). Every time when the group begins to work with another set of information, a new module will be created [16].

The thinkLet usage in the Group DSS design became better known and often uses because of the following advantages:

- Allows a rapid and compact report of the collaboration patterns, including hardware and software instruments, configuration possibilities and execution scripts for the simplifying process in the decision groups.
- They can be reused.
- Fulfills the moderator role in a decision-making group avoiding the presence of a specialized person for this task.

The data warehouse design process is focusing itself on data adjusting in order to satisfy some decision, tactic or strategic requirements.

In the scientific literature the data warehouse design process is seen as a set of operations focused on data adjustment and technical details needed for implementing the data warehouse in a well known database management systems (DBMS), as for example: ORACLE, MS SQL SERVER, IBM DB2, etc.

The most important data warehouse design approaches are the following:

- Review list approach. This approach was first used by William H. Inmon [8] and it is based on a set of questions and successive operation for designing and implementing a data warehouse.
- Multidimensional data architecture based approach. This approach was first used by Ralph Kimball [10] and it is based on the supposition according to which the majority of DSS oriented on Business Intelligence technologies are based on a multidimensional data structure.

The OLAP applications design and implementation process is focused on identifying and designing multidimensional indicators and reports needed in the decision-making process.

Nowadays the implementation and designing OLAP Application techniques are shaped in frameworks and software framework specific for every DBMS.

DB2 Business Intelligence Solution Framework [2] - provides the necessary procedure and components for Business Intelligence solution development in IBM DB2.

The decision for choosing the methodology or techniques for a DSS development is made by taking into consideration the system complexity.

The table 1 reflects the possibilities of using these methodologies and techniques for designing and creating different DSS types.

Table 1. Methodologies, methods and techniques for development different DSS types.

Methodology, methods or technique / DSS Typology	SDLC	ROMC	Functional Category Analysis	General Process for DSS Development	RAD and XP	Prototyping	Unified Process	UML
Individual DSS	√	√	√	√	√	√	√	√
EIS (Executive Information Systems)	√			√	√	√	√	√
GDSS (Group Decision Support Systems)	√	√		√	√	√	√	√
ES (Expert Systems)	√		√	√	√	√		√
IA (Intelligent Agents)	√		√		√	√		√
GW (groupware)	√	√		√	√	√	√	√
DW (Data Warehouse)			√	√	√	√		√
OLAP		√	√	√	√	√	√	√
DM (Data Mining)			√		√	√		√
HDSS (Hybrid DSS)	√		√		√	√	√	√

From this table we can see that UML, RAD and XP technologies and prototyping are extensively used, being therefore recommended in the DSS development.

In our opinion, the strategies for conceiving and developing DSS can be classified in two large categories:

- Conceiving and developing DSS by the organization ("in-house" development). In this case, exclusively the departments in the organization achieve the development of the DSS. The DSS resulted by the use of these strategies have a high grade of peculiarity being specific to the organization.
  This category of strategies includes:
  - developing DSS specific to the organization, by programming languages. This strategy engages either a general-purpose programming language (GPL), such as C++, PASCAL, BASIC or COBOL, or a fourth-generation language (4GL), such as VISUAL BASIC .NET, C# .NET, VISUAL J# .NET, DELPHI, JAVA or VISUAL C++.
  - developing DSS by DSS generators. The DSS generators are packages of software that allow interactive development of a DSS without the need of programming in a certain language. The best-known and most used category of generators is worksheets like MS Excel, Lotus 1-2-3 or Quattro Pro documents. Practically, many organizational and individual DSS applications use a DSS generator.

- Purchasing and customizing some DSS specific to certain activities in the organization. The strategy to purchase a DSS is recommended in the fields in which there are already DSS specific for the activities of the field, DSS being on the market and the their implementation implies only establishing and introducing some parameters that distinguish the activities in a field by other fields. For example, DSS in fields like medicine (surgery, dentistry and pharmaceutics), agriculture (soil analysis), military, stock market, banking, financial services etc. The selection of DSS can be a very complex process that implies a series of activities and decisions. Sprague Jr. and Watson (1996) developed a multicriterial decision methodology for DSS selection [13].

We consider that the selection strategy for purchasing or making a DSS must be strictly documented. In this context, we recommend to consider some aspects, in addition to the requirements of the decision-makers regarding the functionality of the DSS:

- The organizational conditions for implementing the DSS. This aspect refers to the organizational changes that the implementation of the DSS implies, from the point of view of the affected posts, changes of the organization's diagram, qualification level of the users, changes of information circuits and flows.
- The accommodation degree of hardware infrastructure for DSS implementation. This implies the identification and evaluation of the degree in which the hardware infrastructure must be updated to ensure a good performance of the DSS.
- Feasibility analysis of the DSS implementation. It is very important that the implementation of the DSS will bring benefits (tangible and intangible) to the organization and the costs of the implementation to be justified by these benefits.
- Analysis of risks that can arise by the implementing and the usage of the DSS.

4. CONCLUSIONS

We consider that the process of conception and development the DSS must be based on an integrated approach, which could permit the interconnection of all technologies for the decision support with the transactional system and the external data sources and the information of the organization.

ACKNOWLEDGEMENTS

This paper is part of the research project POSDRU/89/1.5/S/59184 *"Performance and excellence in postdoctoral research within the field of economic sciences in Romania"*, Babeş-Bolyai University, Cluj-Napoca being a partner within the project.

REFERENCES

[1] Arlow, J., Neustadt, I., UML and the Unified Process – Practical object-oriented analysis and design, Addison Wesley, Boston, 2002.

[2] Ballard, C., s.a., Leveraging DB2 Data Warehouse Edition for Business Intelligence, IBM REDBOOKS, http://www.redbooks.ibm.com/redbooks/pdfs/sg247274.pdf, 2006.

[3] Beck, K., Andres, C., Extreme Programming Explained: Embrace Change, Second Edition, Addison Wesley Professional, 2004.

[4] Booch, G., Rumbaugh, J., Jacobson, I., The Unified Modeling Language User Guide, 2nd Edition, Addison Wesley, Boston, 2005.

[5] Briggs, R.O., De Vreede, G.J., Nunamaker Jr., J.F., Collaboration Engineering with ThinkLets to Pursue Sustained Success with Group Support Systems, Journal of Management Information Systems, vol. 19, nr. 4, pag. 31-64, Spring 2003.

[6] Briggs, R.O., de Vreede, G.J., Nunamaker Jr., J.F., David, T.H., ThinkLets: achieving predictable, repeatable patterns of group interaction with group support systems. In: Hawaii International Conference on System Sciences. IEEE Computer Society Press, Los Altos, 2001.

[7] Gorry, G.A., Scott Morton, M.S., A framework for Management Information Systems, Sloan Management Review, Vol.13, No.1, 1971.

[8] Inmon, W.H., Building the Data Warehouse, 4th Edition, Wiley Publishing Inc., Indianapolis, 2005.

[9] Jacobson, I., Booch, G., Rumbaugh, J., Unified Software Development Process, 1999.

[10] Kimball, R., The Data Warehouse Toolkit, Wiley, New York, 1996.

[11] Marakas, G.M., Decision Support Systems in the 21st century, 2nd edition, Prentice Hall, New Jersey, 2003.

[12] O'Brien, J., Marakas, G., Management Information Systems, 10th Edition, McGraw-Hill/Irwin, 2010.

[13] Sprague, R.H., Jr., Watson, H.J., Decision Support for Management, Prentice Hall, New Jersey, 1996.

[14] Turban, E., Aronson, J.E., Decision Support Systems and Intelligent Systems, 6th edition, Prentice Hall, New Jersey, 2001.

[15] Turban, E., Sharda, R., Delen, D., King, D., Business Intelligence, 2nd Edition, Prentice Hall, 2010.

[16] Zamfirescu, C.B., Filip, F.G., Metode de foresight în identificarea riscurilor şi asistearea gestionării lor cu calculatorul, colecţia "Fenomene şi procese cu risc major la scară naţională", Editura Academiei Române, Bucureşti, 2004.