A Unified Framework for Decision Support Systems Development

Claudiu Brândaş

University of the West, Faculty of Economic Studies

16, J.H. Pestalozzi, 300115, Timişoara, România

claudiu.brandas@fse.uvt.ro

Abstract: The research of DSS (Decision Support Systems), mainly in the last twenty years resulted in the appearance of new technologies and concepts regarding storing, processing and analyzing data and information necessary for the decision-making process. Consequently in the DSS landscape, the technology of ”Data Warehouse”, OLAP (“On-Line Analytical Processing”) applications, “Data Mining” techniques and artificial intelligence technologies (expert systems and intelligent agents) are present. In this respect DSS development is a complex process, which needs an adequate methodology or framework, able to manage different tools and platforms in order to fulfill management requirements. The DSS development process should be treated as a unified and iterative set of activities and operations. The new tools based on Unified Process (UP) methodology and UML (Unified Modeling Language) seem to be appropriate for DSS development using prototyping and RAD (Rapid Application Development) techniques. This paper presents state-of-the-art DSS development methodology and framework using Unified Process, UML and DSS-UNIDEF.

Keywords: Decision Support Systems, Unified Process, UML, Prototyping, DSS Tools, DSS Development, unified framework.

Claudiu Brândaş is Lecturer at the University of the West Timisoara, Faculty of Economic Studies, Department of Business Information Systems and Statistics. He earned his PhD from "Babes-Bolyai" University of Cluj-Napoca, the Faculty of Economics in Decision Support Systems conception and design. Currently, his research interests include DSS (Decision Support System), Business Intelligence, Business Information Systems Analysis and Design, Business Process Modeling, and Information Systems Control and Audit.

1. Introduction

Since it was first established in 1971 by Gorry and Scott Morton, the Decision Support System (DSS) concept has known a very dynamic evolution, reflected in its architecture as well as in the technologies derived from it.

Today there are significant improvements in the decision support technologies, concerning data storage volumes, data processing as well as data extraction, visualization, and communication. The fast evolution of wireless communication, Internet and WEB technology revealed new directions in the design and development of the DSS.

The structure of decision support environment became very complex due to new generation of Business Intelligence applications and technologies like Data Warehouse, OLAP (On Line Analytical Processing) and Data Mining. These technologies enrich the organization’s business intelligence process, in order to achieve new stages in the Decision Support System design and implementation.

In addition, the DSS design and development process must be based on an integrated approach, which allows the interconnection of all decision support technologies with the organization’s transactional systems and external data sources.

DSS developers must use and integrate various technologies and tools in order to cover decision-making process requirements based on business intelligence process. In this respect DSS development is a complex process, which needs an adequate methodology or framework, able to manage different tools and platforms in order to fulfill management requirements.

The DSS development process should be treated as a unified and iterative set of activities and operations. The new tools based on Unified Process (UP) methodology and UML (Unified Modeling Language) seem to be appropriate for DSS development using prototyping and RAD (Rapid Application Development) techniques. UP aims to incrementally build robust system architecture [8].

Modeling DSS architecture with UML (Unified Modeling Language) using the UP as main methodology, helps the DSS developers to rapidly construct and implement an accurate and scalable DSS. The UP and UML offers appropriate framework for modern DSS developing.

The paper is organized as follows: section two reviews several DSS development strategies; section three presents the alternatives for DSS development using Unified Process methodology and UML language, section 4 explains a unified framework (DSS-UNIDEF) for DSS development, which is based on a simplified form of Unified Process methodology, UML language and prototyping. These methodologies and languages are used in order to create and develop a DSS in an integrated manner. In section 5 the DSS-UNIDEF framework is applied in the development of a unified DSS model, based on rules and OLAP for budget management. Finally, section 6 contains the conclusions to the above mentioned items.

2. Strategies for DSS Development

According to Marakas (2003), there are two strategies for developing DSS [17]: (1) developing organization's customized DSS, through programming languages; (2) developing DSS through DSS generators. In addition to these two strategies, many organizations purchase custom-made DSS for certain activities which are adapted and customized according to the organization requirements.

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

I. Designing and developing DSS in the organization ("in-house" development). In this case, exclusively the departments in the organization achieve the development of the DSS. The DSS resulted is exclusive and specific for organization.

This strategy includes:

a. *developing a specific DSS for the organization through 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++;

b. *developing a DSS through DSS generators.* The DSS generators are software packages which allow interactive development of a DSS without the need of programming in a certain language. The best-known and most used categories of generators are worksheets like MS Excel, Lotus 1-2-3 or Quattro Pro documents. Most of the organizations use DSS generator for DSS applications.

II. Purchasing and customizing DSS software for specific activities in the organization. An organization should adopt this strategy as long as there is already DSS software for specific activities. So that implementation implies only the establishment and the introduction of some specific parameters. For example, DSS in fields like medicine (surgery, dentistry and pharmaceutics), agriculture (soil analysis), military, stock market, banking, financial services, etc. The process of DSS selection can be a very complex involving a series of activities and decisions. Sprague Jr. and Watson (1996) developed a multicriterial decision methodology for DSS selection [22].

We consider the selection of make or buy for a DSS must be strictly documented. Moreover some additional requirements besides those of decision makers should be also considered:

- *The organizational environment in implementing the DSS,* refers to organizational changes the implementation of the DSS implies. For example the affected workplace (individuals), changes of the organization's diagram, qualification level of the users, changes of information circuits and flows.
- *Hardware infrastructure degree of compatibility for DSS implementation.* This process implies identification and evaluation of changes that have to be made to hardware infrastructure in order to ensure a good performance of the DSS.
- *Feasibility analysis of the DSS implementation.* It is very important for the DSS implementation to bring benefits (tangible and intangible) for organization and the costs of the implementation to be justified by these benefits.
- *Analysis of risks bundled with DSS implementation and usage.*

Traditional development of information systems based on *Systems Development Life Cycle (SDLC)*, was the first concept in DSS development and implementation. SDLC can be seen as founder concept of 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 using SDLC is not the best option because of its rigidity and the huge volume of project documentation.

One alternative to the SDLC is the *ROMC analysis* [23]. 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 system analyst will investigate and create models for the existing representations, 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. This method reveals the necessary functions for designing of DSS. These functions are selected from an existing 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.

Another highly used systems development method is the *RAD (Rapid Application Development)*. This method is characterized by a rapid and iterative development of the information systems, using prototyping and CASE (Computer Aided Systems/Software Engineering) tools. 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 system design is made with the direct contribution of end users, ensuring a better system evaluation, before it 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 real time.
- 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 J Builder, etc.

In the 90's, the *XP (Extreme Programming) methodology* was created by Kent Beck, which was developed during several software projects [5]. This technology stands for a rapid and successive development of information system; realized by a direct implementation in production and with 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 and semi-structured nature of 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 with 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 GDSS (Group Decision Support Systems) is possible due to the iteration technique, prototyping and a permanent contact with end users (decision makers).

The use of *prototyping* is the most common method for a rapid development of information systems. It can be found in today's almost all techniques and methods used to develop information systems. This method is known under the name of *iterative design* or *evolutionary development* of information systems [24].

Turban and Aronson [24] argue that most DSS are developed through the prototyping process (technique). They highlighted that "… the prototyping development technique aims to build a DSS in a series of short steps with immediate feedback from users (managers) in order to ensure an accurate and complete system development. Therefore, DSS tools must be flexible to permit changes quickly and easily. The DSS development is focused on a specific, dynamic and complex activity, which requests a series of updates and tests. This may be one of the reasons why the prototyping method has become so popular in the last years." [24]. In the scientific literature one can find two concepts related to the DSS development through prototyping: to create the system based on a *throwaway prototype*; to create the system based on an *evolutionary prototype*.

The DSS complexity imposes an integrated and rapid design process, oriented towards reprocessing the system's components. The answer to this challenge can be found in the *Unified Process (UP) Methodology and UML (Unified Modeling Language).*

3. Unified Process Methodology and UML in the DSS Development Process

Researchers assert the need for new business paradigms to drive a unified approach to development of new active decision support capabilities [20]. Jacobson, Booch and Rumbaugh introduced in 1999 the *Unified Software Development Process (USDP)* as a software engineering process standard [14]. It is commonly referred to as the *Unified Process or UP*. This methodology consists of cycles that may repeat over the life of systems development. UP have three basic axioms [4]:

- *Use case and risk driven*. UP employee Use Cases to capturing users requirements and predicating software construction on the analysis of risk.
- *Architecture centric*. UP approach in developing software systems is to develop and evolve in robust system architecture. Architecture describes the strategic aspects of how the system is broken down into components, and how those components interact and are deployed on hardware.
- *Iterative and incremental*. The iterative aspect of UP means that we break the project into small subprojects (the iterations) that deliver system functionality in chunks, or increments, leading to a fully functional system.

There are several commercial variants of UP available. The most widely used commercial variant is *RUP (Rational Unified Process)*. This product supplies all standards, tools, etc. that are not included in UP and that one would otherwise have to provide oneself. Both UP and RUP are modeling "*the who, when and what"* of the software development process, however they do so very slightly differently.

UP have four main phases, each of them having one iteration workflow:

- *Inception* - creates a vision about the project;
- *Elaboration* - plans all the activities and resources, defines the characteristics and designs the system architecture;
- *Construction* - actual construction of the product, performing several increment iterations;
- *Transition* - product delivery to the user.

In the UP each iteration has a work flow composed from five steps:

- *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.

For modeling process and visual representation UP uses UML (Unified Modeling Language).

The unified modeling language (UML) embodies a visual modeling standard 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.

The DSS development through unified development methodologies is very poor represented in the scientific literature. The most well known approaches are used in modeling and implementation of data warehouses and OLAP applications. Prat et all, perform a study related to the usage of the UML in the modeling and design process of data warehouses [19].

A significant contribution related to the development of DSS, based on techniques and meta-models in a unified approach, is granted to OMG Group.

Using UML in DSS development, design and implementation ensures:

- a better detection of the decision making process requirements;
- a better understanding of the interaction between the decision factors, external environment, existing software and other elements;
- a better and faster communication between the DSS functions;
- a standard for the development process and a pattern for the decision making process (decision making process for the financial accounting activities, decision making process for

marketing activities);

- an increase in 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.

Modeling DSS with UML (Unified Modeling Language) and using the UP as main methodology, DSS developers are able to construct and to implement rapidly, accurate and scalable the DSS. The UP and UML offers appropriate activities and modeling language for developing modern DSS.

DSS development based on UP and UML can be easily implemented using Prototyping. The research in using UP in DSS development process is very poor. Most of it refers only on using UML to model DSS development process.

4. DSS Unified Development Framework

Using a simplified form of the Unified Process Methodology, Unified Modeling Language (UML) and prototyping we have develop a framework for conception, construction and implementation of DSS.

The *DSS Unified Development Framework of the DSS*, shortly *DSS-UNIDEF* (*D*ecision *S*upport *S*ystems – *Uni*fied *De*velopment *F*ramework), represents a combination of a methodology (the simplified form of the Unified Process methodology), activities, processes, visual modeling language (UML), meta-models, models, tools and techniques (prototyping) used for the conception, construction and implementation of

The DSS-UNIDEF is structured on three layers (figure 1) [7].

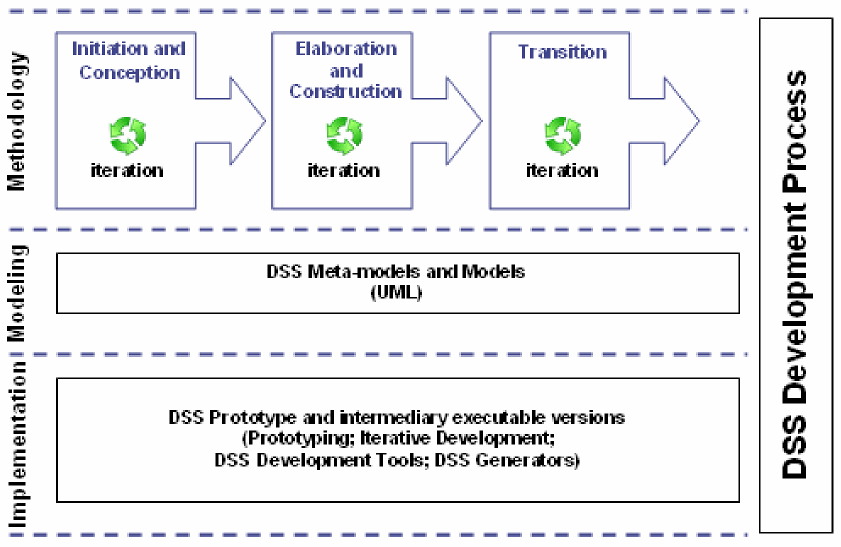

Methodology
Initiation and Conception
iteration
Elaboration and Construction
iteration
Transition
iteration
Modeling
DSS Meta-models and Models (UML)
Implementation
DSS Prototype and intermediary executable versions (Prototyping; Iterative Development; DSS Development Tools; DSS Generators)
DSS Development Process

Figure 1. DSS-UNIDEF layers.

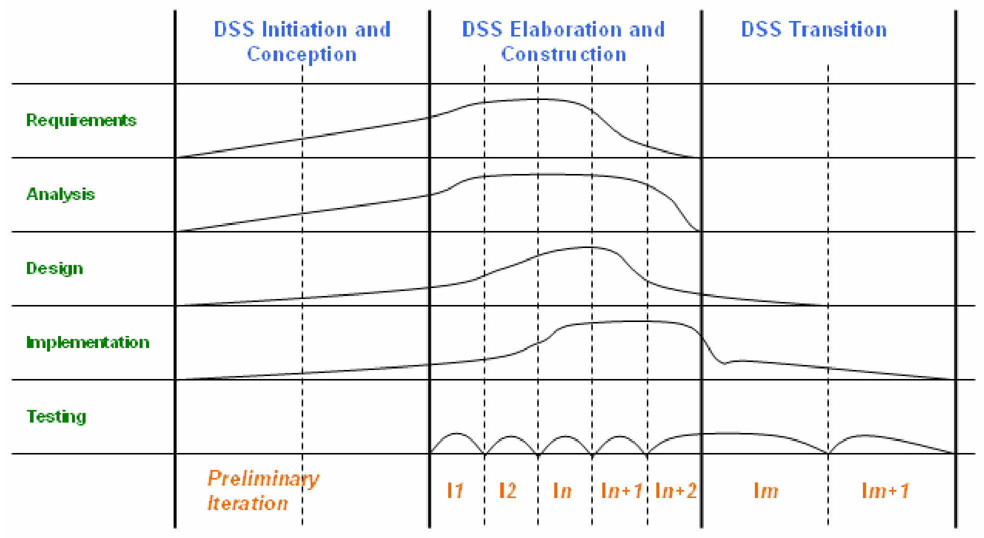
First Layer – The Methodology

The first layer represents the Unified Process methodology (simplified form). As a particularity of this level, we merged the elaboration and construction phases, in order to combine the executable architecture with the intermediary executable versions. This way the development cycle of the system can be reduced. Furthermore, the methodology's steps, matching the framework, will be explained. The three stages of the methodology are:

- *DSS Initiation and Conception.*
- *DSS Elaboration and Construction*
- *DSS Transition*

Concerning the three stages, the development process of DSS follows an iterative workflow: *Requirements* → *Analysis* → *Design* → *Implementation* → *Testing.*

Each iterative workflow phase has a different effect on the system. Figure 2 shows the effect of each workflow phase on the system.



Adapted from JACOBSON,1999

Figure 2. Implication level of iterative workflow phases.

We can notice that in the *initiation and conception stage of DSS*, the most important function is attributed to system *requirements and analysis of the process*. In this stage, the main goal is to *"get the project off the ground"* and the draft (throwaway prototype) of the system is created, as well as the analysis of success risks. The design and implementation phases are less present at this point, while these are used only in the throwaway prototype design. The workflow of the process, at this stage is restricted to few preliminary iterations, merely only two iterations are advisable.

In the second stage, *elaboration and construction*, the decision-making process requirements and analysis have a critical importance. The requirements for the decision-making process are more complex than those necessary for individuals (users) using transactional systems or other applications. This way the requirements and analysis for decision-making process are more present in the elaboration and construction stage. Design and implementation have a major role because both of them open the possibility to elaborate the executable architecture of DSS and intermediary executable versions. The testing phase is included while it is used to test the intermediary versions. Each process workflow has a major contribution because at the end of each flow a new and better version is released.

The transition stage of the DSS includes system transfer to users. The most important step of this stage is *testing*. The executable versions which are tested in this step, have been already „alpha" tested, the „beta" testing has already begun, and in this testing phase the „beta" version will be complete, resulting in a final executable and stable version. From the process workflow scheme presented earlier, some elements are used but only in a reduced scale. These elements are design and implementation. Due to their high level of complexity, the DSS can endure changes even in the testing phase. Therefore in this stage the integration of the design and implementation activities is necessary.

Second Level - The Modeling

The modeling layer uses the unified modeling language (UML). Using UML diagrams and symbols the meta-models and DSS models are represented.

The Third level – The Implementation

The implementation layer contains the prototyping technique and tools for building DSS, such as DSS generators.

The milestones and deliverables of this framework are presented in table 1:

Table 1: The milestones and deliverables of DSS-UNIDEF.

DSS Development Stages	Milestones	Deliverables
♦ **Initiation and conception**	♦ The scope of the project was established and defined ♦ The base requirements of the DSS have been identified and analyzed ♦ The system's structure and collaborations have been established	➢ Decision-making requirements model ➢ Decision-making process model ➢ System collaborations model ➢ General use cases of decision-making process
	♦ The risks of the decision making process have been detected and assessed	➢ Decision-making process risks model
	♦ The feasibility analysis of the process has been performed	➢ The model of feasibility analysis to determine the development and/or implementation of the DSS.
	♦ The initial project plan has been developed	➢ Initial project plan.
	♦ The throwaway prototype has been finalized	➢ Throwaway Prototype versions
♦ **Elaboration and construction**	♦ DSS structural models have been elaborated	➢ Objects and classes models ➢ Data model ➢ System components and packages model ➢ DSS deployment model
	♦ DSS behavioral models have been elaborated	➢ DSS use cases and activities model ➢ DSS interactions and communication model
	♦ The executable-base version of the system has been finalized	➢ Executable-base version
	♦ The users manual has been elaborated	➢ Users manuals
	♦ The beta testing has been completed	➢ Executable version – final release, prepared for transition
♦ **Transition**	♦ Users training plan was elaborated	➢ Users training model
	♦ The conditions necessary for the system installation and integration have been completed	➢ System deployment and integration model
	♦ The installation and integration has been completed ♦ The requests for the maintenance and support plan have been settled	➢ Support and maintenance model

For a better control we have created a *control flow of framework*. This flow allows:

- Stage, phase and activities monitoring;
- Milestones and deliverables for each stages can be supervised;

The DSS-UNIDEF has been developed firstly as a guide and as instrument for conception, development and implementation of DSS and secondly as a harmonized pool where the UML and unified system approach can work together.

The framework architecture is open and will be further developed and customized for the upcoming DSS applications.

5. A Unified Model of DSS Based on Rules and OLAP for Budget Management

Budget and cost management is a very complex activity, which requires data and information from all levels of organization's information system. The *main objectives* of this activity are *planning and controlling*. The decision which has to be made by managers will have a major impact on the whole organization and its position on the market.

In order to fulfill the management needs, *we have developed, using the DSS–UNIDEF framework, a DSS model for the budget and expense management called „B-Admin DSS".*

The main functions of this system are:

- *Expense collection, consolidation and allocation*:
    - data import from financial accounting information system. The data gathered from this system are exported in Excel. Afterwards this information is imported in the B-Admin DSS;
    - data gathered from all centers is consolidated in one common database;
    - indirect expense allocation based on allocation rules;
    - income and expense reports display.
- *Multidimensional reporting based on OLAP cubes*. In order to complete a multidimensional analysis of the processed and centralized data, received from the previously mentioned module, we have implemented an OLAP cube using the special features available in the Business Intelligence component of MS SQL SERVER 2005. The OLAP cube allows managers in an organization to visualize a series of reports having a dynamic and multidimensional structure. Therefore income and expense reports can be presented in a multidimensional structure, being assigned to cost centers, activities, years, months or quarters.

We consider this system to be a Hybrid Decision Support System, which combines the elements and functions of EIS (Executive Information Systems) with the OLAP technology.

Applying the DSS-UNIDEF framework, we can identify decision-making requirements more accurately and we can develop more rapidly (prototyping) the system model in a unified approach based on UML language.

6. Conclusions

Current frameworks and methodologies for DSS Development must be able to integrate Business Process with new DSS applications like Business Intelligence technologies. In this respect DSS developers must have a unified view of DSS building and implementation. In our opinion UP, UML, Prototyping and DSS Tools / DSS Generators are the most recommended for current DSS development and implementation.

DSS models elaborated with UML can be rapidly integrated with other business functions in the new MDA (Model-Driven Architecture). With UML, DSS developers can create complex decision models and databases (Data warehouse, OLAP and Data Mining Applications). Moreover UML offer a high level of component reusability.

In order to create and develop a DSS in an integrated manner based on activities workflow which will be able to identify, modeling and implement accurately the decision-making requirements, *we recommend the usage of the DSS-UNIDEF framework*. This framework should be used as a guide, which will allow a realistic and well documented system capable to meet most user requirements. It can be a basic foundation for the development of flexible, accurate and reusable DSS.

REFERENCES

1. ALBIN, S.T., **The Art of Software Architecture: Design Methods and Techniques**, John Wiley & Sons, New York, 2003.

2. ALTER, S.L., **Decision Support Systems: Current Practices and Continuing Challenges**, Addison-Wesley, 1980.

3. ARINZA, B., A, **Contingency Model of DSS Development Methodology**, Journal of MIS, Summer, 1991.

4. ARLOW, J., NEUSTADT, I., **UML and the Unified Process – Practical Object-oriented Analysis and Design**, Addison Wesley, Boston, 2002.

5. BECK, K., ANDRES, C., **Extreme Programming Explained: Embrace Change**, 2nd Edition, Addison Wesley Professional, 2004.

6. BOOCH, G., RUMBAUGH, J., JACOBSON, I., **The Unified Modeling Language User Guide**, 2nd Edition, Addison Wesley, Boston, 2005.

7. BRÂNDAŞ, C., **Conceptual Framework for DSS Development Based on UP and UML**, Annals of the Tiberiu Popoviciu Seminar, Supplement International Workshop in Collaborative Systems, Volume 4, Cluj-Napoca, 2006, pp 38 – 47.

8. BRÂNDAŞ, C., **Unified Approach in the DSS Development Process**, Economy Informatics Review, 11(1), INFOREC (Ed), Bucureşti, 2007, pp. 98-102.

9. CHANG, LC, TU, Y.M., **Attempt to Integrate System Dynamics and UML in Business Process Modeling**, Proceedings of 21th International Conference of System Dynamics Society, 2004.

10. EOM, B.S., **Decision Support Systems Research: Current State and Trend***s*, Industrial Management & Data Systems, 99(5), MCB University Press (Ed), 1999, pp. 213-220.

11. FILIP, F.G., **Sisteme suport pentru decizii**, 2nd Edition, Editura Tehnică, Bucureşti, 2007.

12. GACHET, A., **A Framework for Developing Distributed Cooperative Decision Support Systems-Inception Phase**, Proceedings of the 4th Informing Science Conference, June 19-22 Kraków, Poland, 2001.

13. GORRY, G.A., SCOTT MORTON, M.S., **A Framework for Management Information Systems**, Sloan Management Review, 13(1), 1971.

14. JACOBSON, I., BOOCH, G., RUMBAUGH, J., **Unified Software Development Process**, Addison-Wesley, Boston, 1999.

15. KEEN, P.G.W., **Decision Support Systems: A Research Perspective**, in Decision Support Systems: Issues and Challenges, Pergamon Press, 1981.

16. MALLACH, E.G., **Decision Support and Data Warehouse Systems**, Irwin McGraw-Hill, Boston, 2000.

17. MARAKAS, G.M., **Decision Support Systems in the 21st Century**, 2nd Edition, Prentice Hall, New Jersey, 2003.

18. PETERS, J.F., PEDRYCZ, W., **Software Engineering: An Engineering Approach**, John Wiley & Sons Inc, 2000.

19. PRAT, N., AKOKA, J., COMYN-WATTIAU, I., **A UML-based Data Ware House Design Method**, Elsevier Science, Decision Support Systems, 2005.

20. RING, J., **Intelligent Enterprises**, INCOSE INSIGHT, 6(2), 2004.

21. SCHACH, S.R., **Object-Oriented and Classical Software Engineering**, 5th ed., McGraw-Hill Companies Inc, 2002.

22. SPRAGUE, JR., R.H., WATSON, H.J., **Decision Support for Management**, Prentice Hall, New Jersey, 1996.

23. SPRAGUE, R. H., AND CARLSON, E. D., **Building Effective Decision Support Systems**, Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

24. TURBAN, E., ARONSON, J.E., **Decision Support Systems and Intelligent Systems**, Prentice Hall, New Jersey, 2001.

25. WATSON, J.H., HOUDESHEL, G., RAINER, R.K. Jr., **Building Executive Information Systems and Other Decision Support Applications**, John Wiley & Sons Inc, 1997.