Contributions to Conception, Design and Development of Decision Support Systems

PhD THESIS
- SUMMARY -

Scientific Advisor:
Professor Ştefan I. NÎŢCHI, PhD

Phd Candidate:
Claudiu BRÂNDAŞ

Cluj-Napoca
2007
## CONTENTS

### Introduction
- Motivation and Objective
- Chapters Summary
- Contributions
- Thesis Structure Diagram

### 1. Introduction in the DSS theory and practice
- 1.1 Decision and decision making
  - 1.1.1 Decision making characteristics and structure
    - 1.1.1.1 Decision
    - 1.1.1.2 Decision Makers
    - 1.1.1.3 Decision Making
  - 1.1.2 Decision Models
  - 1.1.3 Decision Makers Typology
  - 1.1.4 Decision Typology
- 1.2 The definition and features of DSS
  - 1.2.1 Gorry and Morton framework for decision support
  - 1.2.2 Decision Support Systems Concept
  - 1.2.3 Decision Support Systems principles and theory
  - 1.2.4 Decision Support Systems Categories
    - 1.2.4.1 Group Decision Support Systems (GDSS)
    - 1.2.4.2 Executive Information Systems (EIS)
    - 1.2.4.3 Intelligent Support Systems
  - 1.2.5 DSS Classification
  - 1.2.6 The benefits of using DSS in the decision making

### 2. DSS Architecture
- 2.1 DSS General Architecture
- 2.2 The Data Management Subsystem (DMS)
- 2.3 The Model Management Subsystem (MMS)
- 2.4 The Knowledge-Based Management Subsystem (KBMS)
- 2.5 The User Interface (Dialog) Subsystem (UIS)
- 2.6 DSS Communication technologies
  - 2.6.1 Client/Server Technology
  - 2.6.2 The Internet
- 2.7 DSS Hardware Infrastructure

### 3. Business Intelligence Support Systems
- 3.1 BI definition and characteristic
- 3.2 Hybrid Support Systems (HSS)
- 3.3 Data Warehouse
  - 3.3.1 Concept and importance
  - 3.3.2 Data Warehouse Architecture
3.3.3 Hybrid Support Systems based on Data Warehouse

3.4 OLAP
   3.4.1 Definition and features
   3.4.2 OLAP Tools
   3.4.3 OLAP processing language
   3.4.4 Hybrid Support Systems based on OLAP

3.5 Data mining
   3.5.1 Concept and features
   3.5.2 Operations and Techniques
   3.5.3 Hybrid Support Systems based on Data mining

4. Methodologies, methods and techniques for DSS Development
   4.1 Strategies for DSS Conception and Development
   4.2 Methodologies for DSS Development
      4.2.1 Traditional Systems Development Lifecycle approach
      4.2.2 ROMC Analysis
      4.2.3 Functional Category Analysis
      4.2.4 General Process for DSS Development
   4.3 Rapid and iterative DSS Development through Prototyping
      4.3.1 RAD and XP Methodologies
      4.3.1 Prototyping
   4.4 Unified Process Methodology and UML for DSS Development
      4.4.1 Unified Process (UP) Methodology
      4.4.2 Unified Modeling Language (UML)
         4.4.2.1 Structural Diagrams
         4.4.2.2 Behavioral Diagrams
         4.4.2.3 Interaction Diagrams
   4.5 Group Decision Support Systems Development Particularities
   4.6 DSS Development Tools
      4.6.1 DSS Primary Development Tools
      4.6.2 DSS Generators
      4.6.3 Specific DSS Applications
   4.7 DSS Development Project Management
      4.7.1 Methods and Techniques for DSS Development Project Management
   4.8 DSS Development particularities
      4.8.1 Data warehouse Design and Implementation
      4.8.2 OLAP Applications Design

5. DSS Implementation and Integration
   5.1 The importance of efficient implementation strategy
      5.1.1 DSS Implementation characteristics
5.1.2 Implementation Strategy
5.2 Implementation Factors
5.3 Implementation Models
5.4 DSS Implementation Risks
5.5 DSS Integration
   5.5.1 Characteristics of DSS integration process
   5.5.2 Integration Models
5.6 The difficulties of implementation and integration process
5.7 DSS Implementation success evaluation

6. DSS Unified Development Framework – DSS-UNIDEF
   6.1 DSS-UNIDEF Framework phases and contents
      6.1.1 General phases and features
      6.1.2 DSS Initiation and Conception
         6.1.2.1 Identification and Modeling Decision Making needs
         6.1.2.2 Identifying the Decision Making Risks
         6.1.2.3 DSS Development and/or implementation feasibility analysis
         6.1.2.4 Initial Project Plan Elaboration
         6.1.2.5 Prototype Model
      6.1.3 Elaboration and Construction of the System
         6.1.3.1 DSS Structural Modeling
         6.1.3.2 DSS Behavioral Modeling
         6.1.3.3 Elaboration of System Based-version
         6.1.3.4 Elaboration of Users Manuals
         6.1.3.5 Beta Testing
         6.1.3.6 Finalizing the system based-version
      6.1.4 System Transition
   6.2 Framework’s Control Flow
   6.3 Conclusions

7. DSS Unified Model based on Rules and OLAP for Management by Budgets
   7.1 General Context of the DSS Conception and Development
      7.1.1 Management by Budgets - general concepts
      7.1.2 Decision Making Requirements for “County Forestry Department” Budgets and Costs Management
      7.1.3 The Structure of Centers and Activities
   7.2 The Architecture and DSS Execution
      7.2.1 System Architecture
      7.2.2 The Module for collecting, centralization and distribution of expenses
      7.2.3 The Multidimensional Data Analysis and Presentation Function
7.2.4 The Key Performance Indicators Presentation Function
7.3 System Conception, Modeling, Design and implementation

Conclusions

Bibliography

Appendices

Keywords: Decision Support Systems, Decision Making, Conception and Development of Decision Support Systems, Unified Process Methodology, UML, Prototyping, Business Intelligence, OLAP, Data Warehouse, Data Mining, DSS Architecture, DSS implementation and integration, Management by Budgets.
In a competition economy, the success of a company depends mostly on the quality of the decisions made by its managers. Together with the development of the information systems, decision-making involves, more and more, the need of a larger volume of information and a complex process of analyzing and synthesizing this information. This capacity of collecting, processing and analyzing the information required in the decision making process went beyond the human limits, the use of information technologies being needed to support the decision within the decision making process.

It is highly needed the development and implementation of a Decision Support System (DSS), which is necessary in order to assist the management in an organization which store daily a large and diversified volume of data. In this respect the research of DSS, mainly in the last twenty years resulted in the appearance of new technologies and concepts regarding the storing processing and analyzing data and information necessary for the decision making process. Consequently in the DSS landscape appeared, the technology of the “Data Warehouse”, the OLAP (“On-Line Analytical Processing”) applications, the “Data Mining” techniques and the artificial intelligence technologies (expert systems and the intelligent agents).

The area of the decision support system was extended, and it is today more than an application based on spreadsheets. In our opinion, DSS is, nowadays, a concept that defines any kind of information technology focused on the support of the decision making process.

Concluding, 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.

Our thesis is that there is the possibility of designing and implementing of a framework for conception and development the DSS in a unified manner, based on the Unified Process Methodology, the UML (Unified Modeling Language), the prototyping technique and/or generating tools for the implementation of DSS.

The aim of this research is to analyze the methodologies and techniques for conception, development and implementation of DSS, regarding the identification of new possibilities of conception and development these systems.

The paper consists of seven chapters. The first five chapters are dedicated to the analysis of the concept of decision support system and of the methodologies, methods and the techniques of conception, elaboration, implementation and integration of them. The chapters 6 and 7 consist of our own contributions, regarding the elaboration of the framework for development under a unified manner of DSS and the application of this framework in the development of a model of DSS for the management of the revenue and expenses budget at “The County Forest Department”.

1. Introduction in the DSS theory and practice

Within this chapter, we showed a synthesis of the concepts of the decision making process, decision and DSS, pointing out the theoretic and practical context in which DSS have appeared and developed, and the role of these systems within the decision making process of an organization.
The **decision** is a knowledge process of choosing among several alternatives\(^1\). Each decision is characterized by a **definition (formulation/request)**, by a **set of alternatives**, as well as by a **set of decision criteria**.

A major element for the decision making process is represented by the **decision factors (decision makers)** or the **managers**.

In a competition economy, the success of a company depends mostly on the quality of the decisions made by its managers. Together with the development of the information systems, decision-making involves, more and more, the need of a larger volume of information and a complex process of analyzing and synthesizing this information. This capacity of collecting, processing and analyzing the information required in the decision making process went beyond the human limits, the use of information technologies being needed to support the decision within the decision making process.

According to Mintzberg (1980), the managers perform 10 major roles, which can be classified in three important categories: **interpersonal, informational, and decisional**. [MINT80]

To perform these roles, the managers need information, which is supplied efficiently and in a quick manner by the information system.

*All the managerial activities are revolving around the decision making process*. First of all, the manager is a decision maker. The decision makers are distributed, within an organization, on different management levels.

In order to estimate the role of the new information technologies in the decision making process, one must analyze the nature of this process.

**Decision-making** is a process of selection among the various ways of action in order to achieve one or more goals. According with SIMON (1977), *the managerial decision is similar to the complete managerial process* [SIMO77]. Considering the functions of the management, we can say that, the planning, organization and the control involve a range of decisions: What should be done? When? Where? By whom?

The decision making process is very complex, involving managers (individuals or a group), data, information and technology. All these sources are focused towards making decisions based on certain alternatives.

**The representation of the decision making process** was achieved under the shape of various models (patterns), issued by a range of professionals. The best-known model is the one issued by Nobel Prize Laureate - Herber T.A. Simon, research worker in management, at Carnegie-Mellon University. Thus, Simon describes four steps (figure 1) that a manager must go through in making a decision.

---

Starting from SIMON’s model, a series of alternative models and methods were developed. The best-known are:

- Paterson’s process of decision-making [VAN96].
- Kotter’s process model [DEAR83].
- Pounds’ Flowchart (1969) of managerial behavior [POUN69].
- Kepner-Tregoe’s method (1965) [KEPN65].
- Hammond’s model (1998) [HAMM98].
- Cougar’s model (1995, 1996) [COUG95], [COUG96].
- Pokras’ methodology (1989) [POKR89].
- Harrison’s models (1999) [HARR99].

Within the decision-making process, decision makers’ typology varies based on the organizational context and the kind of the decision. Marakas (2003) issued a general model of classifying the decision makers (figure 2)².

---

The decisions can be classified in several categories. According to Gorry and Scott Morton, these can be classified in two dimensions: The **kind of the decision** and its **aim**. [GORR71]

After their **kind**, the decisions can be:

- **Structured decisions** – they are decisions for which there is a well-structured process. This type of decisions can be attached to a software application, although the development of such a system would not be efficient in any situation. More precise, a structured decision is a decision in which the inputs, the outputs and the internal procedures related to the stages of the decision can be specified. Any stage of the decisions for which the above specified are valid, is called the stage of the structured decisions.

- **Non-structured decisions** – those decisions for which the three stages of the decision are not clearly structured. In this case, one cannot specify even one of the three stages: inputs, outputs and internal procedures. This situation can occur because the decision that should be made is either very new or it was not studied carefully enough. In this case, the computers can help users in making these non-structured decisions, the only problem being the large volume of work performed by the users.

- **Semi-structured decisions** – certain aspects of these decisions are somehow structured but we cannot include them in the structured decisions category. This means that only one or two of the three stages are structured. In this case, the computers are of great help because most of the company’s decisions are included in this category.

According to their **aim**, the decisions can be:

- **Strategic decisions** – those decisions that affect the entire organization or most of it for a long period. These decisions affect the objectives and the strategies of the organizations. Usually, these decisions are made at the organization’s management level.

- **Tactic decisions or control decisions of the management** – they affect the way in which a part of the organization will conduct its activities in the future. Usually, these decisions are made in the context the strategic decisions. Ordinarily, these decisions are made at the middle management level of the organization, i.e. the managers situated at a...
level lower those who run the organization and who make the strategic decisions, but can determine a set of actions that are to be made in future.

- **Operational decisions** – this type of decisions affect only the current activities of the organizations without having an effect on future activities. Operational decisions are related only to activities, resources or aims that had been already settled by the strategic and tactical decisions. Ordinarily, these decisions are made at the lower levels of the management or even by the non-managerial personnel of the organization.

The two characteristics – the structure and the aim – are not very co-related, but there is still a relation between them. *Generally, we can say that the operational decisions tend to be structured, and the strategic decisions are unstructured.*

The concept of **Decision Support Systems (DSS)** appeared in the late ‘60s and the early ‘70s, when any person was able to use the computer directly without the help of professionals.

However, for the first time, the term of DSS appears in the works of professors *G. Anthony Gorry* and *Michael S. Scott Morton*, at the beginning of 1971.

Starting with the structure of the decision making process, issued by Simon, and with the management levels of Robert N. Anthony, they both implemented a framework for the decision making process in which the computers are used. Thus, they issued a grid (figure 3) in which, starting with the three management levels, the problems are defined as being **structured** (programmed), **semi-structured** (semi-programmed) and **unstructured** (non-programmed).
[KEEN79], [FILI04], [LEIG86], [KANT92], [KROE92], [HICKS93], [FREN96], [STAI97], [OZ98] and [OBRI99].

In the work hereby, we consider that DSS represent an assembly of software applications designed and integrated for the decision support of management and for the support of knowledge and organizational intelligence.

According to Sprague, the Decision Support Systems are characterized by four basic characteristics: [SPRA80]

- DSS have as object - the poorly structured or non-structured that the managers have to solve.
- DSS combine the using of analytical models and techniques with the traditional functions for the access to system data.
- DSS contain usage facilities, for those who are un-experienced in information technology, based on interactivity.
- DSS have flexibility and adaptability to the changes in the decisional environment.

By analyzing the above-mentioned definitions, the following basic characteristics of a DSS can result:

- DSS are information systems.
- DSS are used for management purposes.
- DSS are used for decision-making.
- DSS are used to support the humans, but not for substituting it totally.
- DSS are used for semi-structured and non-structured decisions.
- DSS contain databases of various categories or sizes.
- DSS contain models for the support of decision-making processes.
- DSS provide support for the organizational knowledge.
- DSS are used for the company’s intelligence.

Even from the very beginning of defining the concept of the DSS there were and than they developed some general principles which characterize this type of systems like The DDM Paradigm (Dialog, Data and Models)³. Sprague and Carlson [SPRA96] formulated this principle. The paradigm DDM assumes that the technology of every DSS must have capabilities in the three domains: dialog, data and models. An efficient decision support system must combine in a balanced way these components in order to be easily used by the non-professional decision factors.

One of the best-known systems for the support of decision groups is the group decision support systems (GDSS)⁴.

By synthesizing the definitions and the characteristics of GDSS, one can say that a GDSS is characterized by the triangle MULTIPLE DECISION MAKER – TECHNOLOGY – DECISION having in its centre the communication activities, cooperation and coordination (figure 4).

---

The **Executive Information System** (EIS) is a particular type of DSS, designed in order to facilitate the information analysis and the monitoring of the performances for all the operations of the organization and to offer the tools that can support the strategic processes of decision-making conducted by the top managers. Turban defines EIS as *a way to provide the decision makers with helpful information and to allow them to focus on the critical data and thus the power to assess the company.* [TURB01]

The development of the applications based on the Artificial Intelligence began to include most of the activity areas.

The **Expert Systems** are a branch of the applied artificial intelligence. They belong to a new branch of the information technologies as Knowledge-Based Systems (KBS).

Elaine Rich and Kevin Knight define SE as being “programs which solve problems that are usually solved by the human experts”. [RICH91].

Eduard Feigenbaum from Stanford University considers that “The expert systems are programs conceived in order to reason skilfully regarding the obligations, which are considered to require a considerable human analysis” [FEIGE84].

In our opinion, the expert systems and the decision support systems can be integrated in four ways:

1. by integrating the expert systems technology within certain components of the decision support systems.
2. development of new components within the decision support systems in order to use the expert systems technology;
3. development of a new architecture of the decision support systems based on the expert systems technology;
4. development of certain intelligent agents to assist the decision-making and the strategic planning based on expert systems technology.

There are a series of applications within the decision support systems for the top management and executive in which the software agents can be integrated. The technology of the intelligent
software agents is still in an early stage, the “great revelations” in this area are to be revealed in the nearest future.

The decision support systems can by classified based on various criteria related to their structure, to their complex usage, on the range of solving the semi-structured problems in the organization and the degree of incorporation within them of the elements of artificial intelligence. The best-known classification is that made by Alter5.

Udo and Guimares (1994) studied the benefits of using the DSS on the 201 corporations in USA [UDO94]. The discovered benefits were:
- the quality of decisions;
- a better communication;
- the reduction of costs;
- high productivity;
- time economy;
- a better satisfaction from both the customer and the employees.

Steven Alter in [ALTE80] issued the classification of the advantages of using DSS, for the first time.

2. DSS Architecture

Having as a destination the decision support in semi-structured and non-structured decisional problems, DSS has a complex structure. According to Alter (1980), the components of a DSS can be classified in seven categories, based on the degree of influence performed by every component on the decision making process. [ALTE80].

According to Turban and Aronson (2001), the DSS structure consists of the data management, the models management, external models, the knowledge-based subsystem, the user interface [TURB01].

The components of a DSS are designed and configured based on the type of the DSS used and the complexity of the decisional problems that it has to assist. These components will be integrated within information system architecture.

The architecture of an information system refers to the way in which the constitutive elements (hardware and software subsystems) are integrated, on the specific types of tasks that are assigned to each component and on the way the constitutive elements are interacting among themselves and with the external environment6.

Generally, DSS architecture should reflect the following elements7:
- The system databases, including the existent databases, that are internal or external to the organization;

---

• The **system model or models**, including the information concerning the data sources and the organizational responsibilities for their maintenance;
• The **system users**, including information about their locations, their jobs, their educational level and any other factor which could affect DSS usage;
• The **software tools through which the users access the database and the models.** Some of these tools, especially those for simple queries on the database, may be supplied even by the database package;
• The **software tools for database and model management**;
• The **hardware platform and the operation system**, on which the databases, the models, the programs are stored and through which the users access the DSS;
• The **communication infrastructure** through which the hardware platforms are interconnected;
• The **level of culture** of the organization that will use the DSS.

In figure 5, we have displayed a conceptual architecture of a DSS.

![Diagram of DSS conceptual architecture](source: [MALL00])

**Fig. 5** The conceptual architecture of a DSS.

In order to present the most important and used components of a DSS (figure 6), we used the classical approach of Turban and Aronson [TURB01]. Thus, a DSS consists of:

• The **data management subsystem (DMS)**. This subsystem includes a database that contains relevant data for the decision making process and is managed by the **database**
management system (DBMS). The data management subsystem can be interconnected with data warehouses.

- The models management subsystem (MMS). This is a software package that includes financial, statistical programs, or other models that offer to the system analytical capabilities. This subsystem also contains the modeling languages for constructing the models.

- The knowledge-based management subsystem (KBMS). This subsystem can be connected to any of the other subsystems or may act as an individual component. It offers artificial intelligence features in order to complete the decision maker.

- The interface subsystem. The user communicates and commands the DSS through this subsystem.

![Fig. 6 The general structure of a DSS.](image)

Most organizations use nowadays the client/server technology in order to access and manage the data, through some terminals that are distributed in the entire organization. The term client/server defines any situation in which an application runs in a shared way on one or more systems of various capacities [MALL00].

DSS can use data belonging to the entire organization, thus being necessary their implementation in client/server communication architecture. One can meet a particular case, which needs compulsory client/server architecture and this is the group decision support system (GDSS).

The present evolution of the information systems and of the communication technologies within the organizations, lead us to the conclusion that the major part of DSS are conceived and implemented on the client/server architecture. We consider that the necessity of using the
client/server architecture within the DSS is also emphasized by the present DBMS, which are exclusively constructed on this architecture. For example, ORACLE, IBM DB2 and MS SQL SERVER are developed and function on client/server architecture.

Nowadays, the Internet is a huge “world wide” source of information. World Wide Web (www) provides a rapid access to information in various areas. Most of this information can be used in the decision making process.

Creating the access of DSS to Internet through a WEB interface may increase its potential both as regarding the volume and the diversity of the data and information it may access and as the dynamic of presenting the information.

Concerning the DSS access to Internet, we consider that one must analyze, thoroughly, the need of this access and the implications, including the risks involved. The explanation of implementing the DSS access to Internet must be documented and approved by the decision makers involved, because such an implementation may affect in a negative manner the integrity, the availability and the accuracy of system data and information.

An important part in the proper functioning of a DSS is represented by the performance of the hardware infrastructure. By infrastructure, we mean the assembly of equipment, devices and installations interconnected in order to make an information system work in an organization. The general components of a hardware infrastructure are as follows: computers (working units, servers, laptops), PDA, printers, scanners, plotters, routers, switches, multimedia equipment, uninterruptible power supplies (UPS). The conception of a DSS must consider the hardware required for the system to function properly.

In our opinion, the factors that influence the structure and the level of the hardware infrastructure’s performances requested by a DSS are as follows:

- The typology of the decision process in which the DSS will be used.
- The volume of data needed for processing and for analyzing.
- The number and location of users.

In order to achieve a fair valuation of the hardware infrastructure needed by a DSS, we propose certain aspects that may be considered:

- Determining the number of users and the degree of their dispersion in the organization.
- The way of updating and access to the DSS resources (interface, reports, database, models base, knowledge base). In this respect, one must determine if the access and the updating must be performed in real time and from different locations.
- The systems and the applications that will communicate with the DSS.
- The data warehouse technology and OLAP applications usage.
- The database management system needed.
- The volume of data and information needed to be processed and analyzed.
- The level of confidentiality of the information within the system.
- The complexity of the queries on the database.
- The complexity of the algorithms used for the model base.
3. Business Intelligence Support Systems

The concept of Business Intelligence within information systems was relatively recently underlain since 1990, although many technologies covered by this concept are rather old.

Business Intelligence can be explained as Organization Intelligence in the context of data, information and knowledge refinement processes.

Synthesizing the most important properties of Business Intelligence, one can say:

- That not only they include the latest and the best decision support technologies, but also they offer predefined and customized solutions for different activities.
- They are focusing on accessing and delivering economic information to decision-makers.
- They include all the information resources needed for supporting decisions, not only the information included in data warehouses.

IBM defines Business Intelligence: “Business Intelligence means using data for making the best decisions for the company. This means accessing, analyzing and discovering new opportunities” [SUEL99].

In our opinion, the most important objectives of Business Intelligence are:

- Collecting and analyzing a large amount of data and information gathered either from operational databases, either from the company’s data warehouses;
- Obtaining forecasts regarding strategic indicators for the organization;
- Combining the Knowledge Management Process with decision processes;
- Making the best out of decision support technologies and providing managers with complex and competitive information.

According to IBM, the structure for a BI system (figure 7) consists of all technologies for supporting decisions as well as the procedures and techniques for managing and combining them in order to maximize the efficiency of the decision making process.

Considering the two approaches, one can say that Business Intelligence is a new vision of the organizational decision process. The BI activities are anchored on information-decision infrastructure provided by DSS combined with other computerized instruments for decision supporting (DW, OLAP, DM, KM and ES).

These activities take part in combining the capacities provided by the information-decision system with the vision and the complex decisional needs of the decision makers. From this point of view, we can say that BI adds value to both decision support systems and the managers of the organization.

Generalizing, we can say that BI has a greater economic connotation linked to managers’ vision and it is focused mainly on the decision process. The technical support for BI is ensured by existing technologies integrated in a Hybrid Support System (HSS) or Hybrid Decision Support Systems (HDSS). This kind of systems forms a real Business Intelligence Support System (BISS).
We think that considering the technology integrated in a BI system, the DSS is a basic component that can be combined with other decision support instruments, and the result is a HSS.

Grouping the BI support technologies by the relationships between them, we propose a conceptual architecture for BI based on DSS (figure 8).

**Hybrid Support Systems (HSS)** represent the systems that are the result of integrating DSS with other tools and technologies for decision supporting in order to maximize the efficiency and efficacy of organizational decision-making process.

The context of BI favors the design, the implementation and the use of HSS.
The best architectures of HSS can be:

- the DSS – Data Warehouses integration: maximizes the efficiency of aggregated organizational data processing and analysis;
- the DSS – OLAP integration: maximizes the efficiency of multidimensional data analysis and extended operation of system models;
- the DSS – Data Mining: maximizes the efficiency of data analysis and the optimization and the extension of system models;
- the DSS – Expert Systems: maximizes the management capacity of data and models as well as the system capacity to manage knowledge;
- the GDSS – Neuronal Networks: allows creating a new structure for knowledge acquisition and storing.

Since 1990, based on developing technologies for data storing, sending and processing there was a set up of aggregating historic and operational data into data structures that allow a unitary data analysis, much faster and less expensive for the organization.

The existing On Line Transaction Processing (OLTP) systems were not capable to meet the new decisional requirements any more. The decision support systems (DSS) in the organization have suffered a change in the DBMS component, by developing the Data Warehouse technology.

The Data Warehouse technology aggregates and integrates many large databases and organizes them in a subject-oriented structure (decisional elements) that supports the decision-makers in an organization or other data-based applications. Aggregation consists of both current operational-level data and historical data. The structure is based both on actual data (the database) and on the rules for calculation, extraction, refining, querying and metadata (data about data).

There are several definitions of the Data Warehouse concept.
The most accepted definition belongs to William Inmon, one of the pioneers of this concept. According to him, the data warehouse is “a collection of subject-oriented, integrated, non-volatile and time-variant data for the management's decision support system” [INMO05].

The Data warehouse architecture can be structured according to several approaches. Thus, Gray and Watson (1998) split the data warehouse in three parts:

- **the actual data warehouse**, that contains data and software instruments for managing data;
- **applications (back-end)** for extracting and taking over the data in the organization’s databases or external sources, consolidating and loading data in the actual data warehouse;
- **client applications (front-end)**, through which the users access and analyze the data in the data warehouse.

A general architecture of the data warehouse is displayed in figure 9:

![Fig. 9 The general architecture of a datawarehouse.](image)

We think that data warehouses fill out or substitute very well the DBMS component of the DSS. In our opinion, the combination between DSS and data warehouses creates a competitive Hybrid Support System.

A way to integrate the two technologies is displayed in figure 10.

The OLAP\(^8\) technology was preceded by a series of instruments that struggled to optimize data access and analysis. One of the most important technologies to access a database is the OLTP technology (**On-Line Transaction Processing**).

The OLAP (**On-Line Analytical Processing**) designates a series of techniques for processing and centralizing data stored in multi-dimensional databases in order to be offered to the decision-makers in a flexible form for analysis.

---

E. F. Codd, who defines five rules that describe an OLAP application [CODD93], introduced the concept of OLAP in 1993. These rules were grouped in a set, called FASMI (Fast Analysis Shared Multidimensional Information).

**Fig. 10** The architecture of a hybrid support system based on data warehouses.

*Considering the DSS, we think that the OLAP technology brings changes both within the DBMS component and the model based management system.*

Although OLAP systems can answer easily to questions like “Who?” or “What?” their main feature is finding the answer to more complex questions like “What if...?” or “Why?”. OLAP allows users to make decisions regarding future actions.

OLAP instruments can be categorized according to the database architecture, which provides data for online analytical processing:

- multidimensional OLAP (MOLAP or MD-OLAP);
- relational OLAP (ROLAP), also called multi-relational OLAP;
- management query environment (MQE), also called hybrid OLAP (HOLAP).

The data analysis process within OLAP applications can only be achieved by using languages that allow operations on cubes. Such languages are SQL, MDX and RISQL.

Using OLAP in the decision process creates a complex architecture of DSS. *We think that through its functions, an OLAP application can be used as an independent instrument or decision support, but also in combination with DSS, leading to the construction of a HSS.*

The OLAP intersects with the DSS in both the DBMS component and the model management one.

In figure 11, we propose a conceptual architecture for a hybrid support system based on OLAP. The bare information storage in a data warehouse does not always bring the benefits that an organization needs, especially considering the implementation of a BI system. We need to extract hidden knowledge within the warehouse in order to be able to fructify its value. As the
quantity of data in the warehouse increases, it becomes more and more difficult if not impossible for the analysts to identify the trends and relationships between data using simple tools for queries and reports.

![Diagram](image.png)

**Fig. 11** The architecture of a hybrid support system based on OLAP.

Not even the usage of the OLAP allows discovering the hidden relationships in a data warehouse. With the help of OLAP technologies, one can see the information that is searched on purpose by the users. This process represents a **verify-oriented analysis**, based on what is already known.

Data mining is one of the best methods to identify new trends and patterns of business out of the multitude of data available. Data mining discovers information in the data warehouse that usual tools for queries and reports cannot distinguish effectively. This process represents a **discovery-oriented analysis**.

Simoudis (1996) stated a widely agreed definition. According to him, **Data Mining is the process, through which one extracts valid information, unknown before, comprehensive and actionable, out of the data warehouse and it is used in making important decision for business.** [SIMO96]

There are four main operations associated with data mining techniques:

- **foresight modeling**;
- **database segmentation**;
- **link analysis**;
- **deviation detection**.

Considering the capacities of data mining, we think that integrating this technology with the DSS creates a high potential in optimizing and updating the models in DSS.

In figure 12, we propose a conceptual architecture of a hybrid system based on data mining.
In our opinion, now, the implementation of HSS in the management of organizations oriented towards BI is very expensive and complex. The complexity and the high price of this kind of implementation come from the heterogeneous technologies and frameworks involved, and the hardware resources needed.

Nevertheless, the organizations can integrate these systems on different levels of complexity and compatibility, so that the cost of implementation will be the lowest. We consider that the success of implementing new technologies for decision support depends largely on the maturity and the structure of the information system of the organization.

4. Methodologies, methods and techniques for DSS Development

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 [GORR71] 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.

Making generalized, historic analysis on the evolution of DSS technologies in terms of time and the evolution of the best-known methodologies for system development, we can observe on a multi-dimensional diagram (figure 13) that these followed a parallel ascending flow.

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 (2003), there are two strategies for developing DSS\footnote{Marakas, G.M., Decision Support Systems in the 21st Century, Prentice Hall, New Jersey, 2003, page 450.}: (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.

\[\text{Fig. 13 The parallel evolution of DSS technologies and the methodologies for developing information systems}\footnote{MDA (Model Driven Architecture) represents one of the latest approaches in information systems design.}.

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

\textbf{I. 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:

\textit{a. 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++.

\textit{b. 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.

II. 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.11

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.**

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** [SPRA82]. 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 [BLAN79]. This method reveals the necessary functions for design of DSS. The functions are selected from 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 [MARA03]:

- **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 [BECK04]. 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\(^\text{12}\).

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 scientific literature one can find two concepts related to the DSS development through prototypes:

- 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\(^\text{13}\). 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:

- 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;

---


\(^{13}\) The base of the Unified Process was established by Ivar Jacobson, Grady Booch, James Rumbaugh in the book “Unified Software Development Process”, 1999.
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 workflow 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;

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\(^\text{14}\).

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\(^\text{15}\). UML offers systems designers a vast thesaurus and a set of rules for conceptual and physical modeling and communication\(^\text{16}\).**

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\(^\text{17}\). 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.).

\(^{14}\) [http://www-306.ibm.com/software/awdtools/rup/]
\(^{17}\) [http://www.sei.cmu.edu/legacy/case/case_whatis.html]
The UML thesaurus comprises a series of diagrams and symbols which can be used to represent the system’s model. UML diagrams (figure 14) 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.

![UML Diagrams (version 2.0)](image)

**Fig. 14** UML Diagrams (version 2.0).

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.

---

18 [http://www.omg.org/gettingstarted/what_is.uml.htm](http://www.omg.org/gettingstarted/what_is.uml.htm)
Collaboration Engineering 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 [BRIG01].

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.

The thinkLet usage in the DSSG 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 [INMO05] and it is based on a set of questions and successive operation for designing and implementing a data warehouse\(^{21}\).

- **Multidimensional data architecture based approach.** This approach was first used by Ralph Kimball [KIMB98] 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\(^{22}\).

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\(^{23}\)* 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.**

<table>
<thead>
<tr>
<th>DSS Typology</th>
<th>SDLC</th>
<th>ROMC</th>
<th>Functional Category Analysis</th>
<th>General Process for DSS Development</th>
<th>RAD and XP</th>
<th>Prototyping</th>
<th>Unified Process</th>
<th>UML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individual DSS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>EIS (Executive Information Systems)</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GDSS (Group Decision Support Systems)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ES (Expert Systems)</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>IA (Intelligent Agents)</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>GW (groupware)</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>DW (Data Warehouse)</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>OLAP</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>DM (Data Mining)</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>HDSS (Hybrid DSS)</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

---


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

5. DSS Implementation and Integration

According to Marakas, implementation is, in fact, the introduction of change [MARA03].

The way of the DSS implementation was made, has a significant impact on the way the users perceive the new system. A poorly prepared implementation which will result in usage difficulties, will generate a negative user perception of the entire system. Therefore it is imperative that the implementing procedures to be designed in way that will prevent the system malfunction.

Implementing a DSS is a very complex activity; it is not just a transactional system collecting, saving, processing and distributing information, a DSS alter especially the decision-making environment. Every change affecting this environment will alter the entire organization. Thus we can conclude that the designing phase of such a system must be very well prepared and it needs a special attention.

In the last 30 years there were designed and developed several implementation strategies.

Cute et all (2000) recommended a few strategies to be used in order to prevent informational system failure, focusing on the process of risk awareness and choosing the right managerial behavior to minimize that risk [CULE00].

Alter (1980) identified four strategy categories for DSS implementation process, each of them having its own set of characteristics and limitations [ALTE80].

It is considered that strategy selection process has a strong impact on the project success. Using a strategy in the implementation process of a DSS enhance but does not guarantee its success.

The most important success factors of a DSS implementation are presented in the figure 15.

![Figure 15 DSS implementation success factors.](source: TURB01)
Both theory and practice provide a series of success models in implementing DSS. In the following we will focus on the most important of them.

The Kolb and Frohman Model (1970)-This model consist in seven steps that need to be followed in order to achieve a successful system implementation. The basic idea of this model states that success is dependent on some action patterns developed between users and analysts.

Alter (1980) identified six scenarios for a successful DSS implementation process. These different scenarios are employed taking into account the users experience level, their responsibility degree and their participation in the designing and creating processes. The implementation risk can be reduced if its factors are known. Therefore one must assess the implementation risk even from the designing phase of a DSS.24

According to Alter there are eight risk factors which can decide the success or failure of a DSS implementation process.

We consider as very important an assessment of the risk factors to be done using risk evaluation techniques like risk analysis surveys.

The necessity of DSS integration with other technologies in order to extract and analyze the information, made the integration process of a DSS into the architecture and infrastructure of organization’s informational system, to receive a greater attention

DSS integration process interconnects the DSS with other technologies and applications of the organization’s informational system ensuring a unitary communication on normal parameters.

According to Marakas (2003) there are two types of DSS integration:

- Functional Integration
- Physical Integration

Turban and Aronson (2001) consider the functional integration as being possible on two levels: (1) between different DSS, (2) in the same DSS.

Most of the DSS integration models are focused on the DSS, expert system and executive system integration in the organization’s system architecture.

Marakas (2003) consider as the most important integration aspect, the capability of a DSS to interconnect globally, to access all the communications channels and to use all the decision mechanisms of an organization. In this context he proposes a general and global model of DSS integration.

The new technologies based on organization portals could have a major role in the functional integration of a DSS. Therefore we propose a functional integration model at the interfaces and DSS access levels using a special form of a portal named decision portal, which can facilitate the collaborative processes from the organization.

In conclusion we consider the decision portal an integrated decision support system interface and a collaborative technology, creating both a unitary environment for the contribution on decision-making process, and a permanent support for the decision factors.

Turban and Aronson (2001) have discovered and synthesized some problems which may occur in the process of DSS integration. The problems are:

- The need for integration.
- Feasibility Analysis.
- Integration’s architecture
- People related problems
- Finding the right and competent programmers
- Attitude problems of the IT department personnel
- DSS development process
- Organizational impact
- Problems related to data structures
- Data related problems
- Connectivity

Klein and Methlie (1995) have developed a framework (table 2), which includes four categories of indicators used to evaluate the success of the DSS.

**Table 2** Framework for DSS implementation success evaluation.

| System performance | ♦ Efficiency and response time  
|                    | ♦ Data entries  
|                    | ♦ Output format  
|                    | ♦ Hardware  
|                    | ♦ Utility  
|                    | ♦ Human – machine interface  
| Task performance   | ♦ Timeframe for decision, alternatives, analysis, quality and participants  
|                    | ♦ Users perception related to trust, satisfaction, utility and understanding  
| Business opportunities | ♦ Development, operating and maintenance costs  
|                      | ♦ Profits related to revenue increase and costs decrease  
|                      | ♦ Value added due to better services, competitive advantage and human know how  
| Evolutionary aspects | ♦ Flexibility degree, changing ability  
|                      | ♦ General functionality of tools  

Source: [KLEI95]

We believe that the implementation and integration process of DSS must be seen as a unified entity beginning from the conception and development phases. Our proposition implies that the success of the DSS has to be evaluated based on analyzing the above mentioned models, factors and criteria.

---

6. DSS Unified Development Framework – DSS-UNIDEF

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 al., performs a study related to the usage of the UML in the modeling and design process of data warehouses [PRAT05].

A significant contribution related to the development of DSS, based on techniques and metamodels in a unified approach, is granted to OMG Group\textsuperscript{26}.

In this chapter we have develop a framework for conception, construction and implementation of DSS, using a simplified form of the Unified Process Methodology and unified modeling language (UML). The framework is implemented in Enterprise Architect 6.5 created by Sparx Systems\textsuperscript{27}. The Enterprise Architect 6.5 is the most complex CASE tools for information system modeling using UML language.

The DSS Unified Development Framework of the DSS, shortly DSS-UNIDEF (Decision Support Systems – Unified Development Framework), represents an combination of a methodology (the simplified form of the Unified Process methodology), several activities, processes, a visual modeling language (UML), meta-models, models, tools and techniques (prototyping) gather and used for the conception, development, construction and implementation of DSS in a unified manner.

The DSS-UNIDEF is structured on three layers\textsuperscript{28}. In the figure 16 can be observed the layers of the framework.

---

\textsuperscript{26} Common Warehouse Metamodel™ (CWM™) Specification, v1.1
\url{http://www.omg.org/technology/documents/formal/cwm.htm}

\textsuperscript{27} \url{http://www.sparxsystems.com.au/}

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. Further, 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 phases has a different effect on the system. Figure 17 shows the effect of each workflow phases on the system.

![Fig. 17 Implication level of iterative workflow phases.](image)

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 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, the 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 using transactional systems or 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. This drawback of the testing phase can be overcome through the integration of the design and implementation elements.

**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.

1. **DSS initiation and conception**

DSS initiation and conception focuses on detecting the necessary requirements of the decision makers, the decision-making process structure, and the risks which could occur during the elaboration of decision-making process models and of the DSS prototype.

In this stage, the most important role is played by identifying and analysis of decision-making process requirements.

This stage includes the following activities (Figure 18):

- **Identifying and modeling the decision making requirements.** The main objective of this stage is to generate the decision-making requirements model, the decision-making process model, the system collaboration model and to determine the general use cases of the decision-making process.

- **Decision-making process risks identification.** This stage has its main goal in identification of the risks that may occur in the decision making process.

- **Feasibility analysis of the DSS development and/or implementation.** Based on the DSS’s requirements and risks, the opportunities and feasibility of the project are analyzed.

- **Elaboration of the initial project plan.** The main goal of this step is the project management. The project management is based on establishing the
project’s plan, resource allocation and set up of control and execution procedures.

- **Throwaway prototype model.** This represents the DSS throwaway prototype.

The milestones and deliverables of this stage are presented in table 3:

**Table 3** The milestones and deliverables of DSS initiation and conception stage.

<table>
<thead>
<tr>
<th>Milestones</th>
<th>Deliverables</th>
</tr>
</thead>
<tbody>
<tr>
<td>♦ The scope of the project was established and defined</td>
<td>Decision-making requirements model</td>
</tr>
<tr>
<td>♦ The base requirements of the DSS have been identified and analyzed</td>
<td>Decision-making process model</td>
</tr>
<tr>
<td>♦ The system’s structure and collaborations have been established</td>
<td>System collaborations model</td>
</tr>
<tr>
<td>♦ The risks of the decision making process have been detected and assessed</td>
<td>General use cases of decision-making process</td>
</tr>
<tr>
<td>♦ The feasibility analysis of the process has been performed</td>
<td>Decision-making process risks model</td>
</tr>
<tr>
<td>♦ The initial project plan has been developed</td>
<td>The model of feasibility analysis to determine the development and/or</td>
</tr>
<tr>
<td>♦ The throwaway prototype has been finalized</td>
<td>implementation of the DSS.</td>
</tr>
<tr>
<td></td>
<td>Initial project plan.</td>
</tr>
<tr>
<td></td>
<td>Throwaway Prototype versions</td>
</tr>
</tbody>
</table>
2. DSS elaboration and construction

The end phase of the initiation and conception stage, the DSS team has to develop the structure and behavior models, to start the set up and to finalize the executable version of the system.

**Objectives:**
- DSS structural models development
- DSS behavioral models development
- Executable version construction and testing
- Users manuals development

Elaboration and construction activities are presented in the figure 19:
- DSS structural modeling
- DSS behavioral modeling
- Elaborating the system executable version
- Elaborating the users manuals
- Beta testing
- Finalizing the system executable version
As related to the process workflow presented above, it can be stated that its main focus is implementation (software building), analysis (modeling the new requirements which have occurred after the system initiation and conception), and design.

The milestones and deliverables of this stage are described in table 4. It can be easily stated that the highlight of this stage is the finalization of the executable version of the system.

**Table 4** The milestones and deliverables of the DSS elaboration and construction.

<table>
<thead>
<tr>
<th>Milestones</th>
<th>Deliverables</th>
</tr>
</thead>
<tbody>
<tr>
<td>♦ DSS structural models have been elaborated</td>
<td>✓ Objects and classes models</td>
</tr>
<tr>
<td></td>
<td>✓ Data model</td>
</tr>
<tr>
<td></td>
<td>✓ System components and packages model</td>
</tr>
<tr>
<td></td>
<td>✓ DSS deployment model</td>
</tr>
</tbody>
</table>
3. DSS transition

The DSS transition stage starts with the final release of the executable version and implies the preparation of the organization’s information system, DSS installation, users training and system integration, support and maintenance.

**Objectives:**
- User’s training and system preparation for the new DSS installation and integration.
- Elaboration of the users training model
- System deployment and integration model
- Elaboration of the maintenance and support plan model

As related to the process workflow presented above, it can be stated that the main focus is implementation (software installation and integration) and testing (last updates of the final version)

The activities of the system transition (figure 20):
- **Users training.** This means the elaboration of a user training model, for which activity diagram will be used.
- **DSS installation and integration.** The result of this activity is the model of system installation and integration. It will be use the deployment diagram, use cases diagram, and component diagram.
- **Elaborating the maintenance and support initial plan.** This activity comprises the plan for support and maintenance. It will be use the activity diagram, use cases diagram, and maintenance diagram.
The milestones and deliverables of the current stage are exposed in table 5. We can notice that installation and finalization of the executable version as well as system deployment in the decisional environment are the main goals of this stage.

Table 5 The milestones and deliverables of the DSS transition stage.

<table>
<thead>
<tr>
<th>Milestones</th>
<th>Deliverables</th>
</tr>
</thead>
<tbody>
<tr>
<td>♦ Users training plan was elaborated</td>
<td>♦ Users training model</td>
</tr>
<tr>
<td>♦ The conditions necessary for the system installation and integration have been completed</td>
<td>♦ System deployment and integration model</td>
</tr>
<tr>
<td>♦ The installation and integration has been completed</td>
<td>♦ Support and maintenance model</td>
</tr>
<tr>
<td>♦ The requests for the maintenance and support plan have been settled</td>
<td></td>
</tr>
</tbody>
</table>

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 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. We can say that our framework has reached the 1.0 version.

7. DSS Unified Model based on Rules and OLAP for Management by Budgets

Budgets and costs management is a very complex activity, which requires data and information from all points of the organization’s information system. The main objectives of this activity are planning and control. The decisions who have to be taken by the managers involved in the budget and cost management could have major impact on the whole organization and its market position.

The information flow on which the manager’s decision related to budget and cost management is provided by the financial accounting information system.

Further, we will present a practical example of a DSS model conception and development, used for budget and cost management in the case of „County Forestry Department”

For a good performance the County Forestry Department need efficient and effective revenue and cost management. A highlight of this case study is that, even if it is a governmental institution, the Forestry Department offers a wide range of goods and services. This is the reason for the Department to be in some sectors, profit center.

Decision making right on time and based on real and reliable information is a major problem, in the context where the information system doesn’t provide, in a centralized and integrated manner, data related to revenue and cost structure and volume. Decisions are taken based on financial data centralized and integrated through a manual process. In this way the data is late and erroneous related to expenses centralization and allocation.

To overcome the above mentioned drawbacks, the manager has created a management strategy based on a real time centralization of income and expenses for all the organization’s subunits in cost/profit centers and activities. The management strategy also comprises the systematization of data and information under a multidimensional format which allows the data to be used and exploit by different point of view. Moreover the manager has chosen a system which will codify the operational activities (on centers and activities) and a system of rules (allocation keys) for the indirect expense allocation. These both systems will automatically collect and allocate the cost on centers and activities.

The management needs related to data to be used in the decision process for budget and expenses management can be described as following:

♦ Design and implementation of a coding system for all the cost/profit centers with the purpose of attribution of codes for each primary document and operation.
♦ Design and implementation of a Decision Support System for budget and expenses management with the objectives:
  o Income and expenses collection and allocation on the level of the whole organization.
  o Integrated reports (multidimensional) for income and expenses.
  o Integrated multidimensional reporting for income and expenses per centers.
Integrated multidimensional reporting for income and expenses per activity.

In order to fulfill the managerial needs, we have created, using the DSS–UNIDEF framework, a DSS model for the budget and expenses management named „B-Admin DSS”. This system has as main functions:

- Expenses collection, centralization and allocation module
- Multidimensional reporting module based on OLAP cubes.

We consider that this system is a hybrid decision support system, which combines the elements and functions of EIS (Executive Information Systems) with the OLAP technology.

By employing the DSS-UNIDEF framework, we can identify the decision-making requirements more accurate and we can elaborate rapidly (prototyping) the system model in a unified approach based on UML models.

The system architecture (figure 21) can be structured from a software and hardware view.

The software side of the system has the following elements:

- Expenses collection, centralization and allocation module
- IIS Web server.
- APACHE Web server.
- MySQL DBMS.

The hardware side of the system comprised the following base elements: CFD server which has a router and web server function, CFD application server; Terminals: CFD control center, districts; Modems; Switch.

According to the system architecture, the users can be grouped in three main categories:

- **Decision maker / Manager**: represents the decision factors (General Manager, Finance Manager, and Chief Engineer) which used the reports and OLAP cubes in order to gather data and information for taking better decisions.
- **The responsible for DSS parameters and expenses allocation**: this is the person who is in charge with the system parameterization (this person has to add into the system the centers, activities, expenses allocation rules, users) and with the monthly generation of the expenses allocation using the system.
- **Primary documents operator**: these are the persons responsible for input into the system the primary documents received from the subdivisions and headquarters of the County Forestry Department. Each entry will receive a code composed form the center’s code and the activity’s code for that entry. This code allocation, allows a direct revenue and cost collection on each center and on each activity.
Expenses collection, centralization and distribution module allows:

- data import from the financial accounting information system. The data gathered from this system are exported in Excel and then further are imported in the B-Admin DSS;
- data centralization from all centers in one common database;
- indirect expenses allocation base on the allocation rules;
- income and expenses reports display.

This module is organized in two sub-modules:

- The sub-module for operating and importing data from the finance and accounting information system module.
- The sub-module for system administration, expenses allocation and reporting.

The sub-module for operating and importing data from the finance and accounting information system module

This sub module is used by the operators from the county forestry department and its subsidiaries. The sub module is employed to import data from the financial and accounting information system of the center to which they are affiliated at.
Each operator access the subsystem’s client interface with a username and password through web browser like Internet Explorer or Mozilla Firefox. The username and password is assigned to a specific unit from the organization (the forestry department headquarters or its subsidiaries). Once the user is logged in he can choose the time span for which he is importing the data and then he would load the Excel file with the income and expenses assigned to the center.

**The sub-module for system administration, expenses allocation and reporting**

This sub module can be accessed only by the system administrator, with a username and password, being his task to start the expenses allocation process. After the data import was completed, the user can see the import status, namely the months and the organization’s units for which the import was made. The process of distributing the indirect costs can be started only after a successful data import, accessing the option “Repartizare” (Allocation) from the “Operatii” (Operation) menu.

In our opinion the data collection, centralization and expense allocation module is a powerful tool used in tracking, allocating and visualizing income and expenses, being an important component in the management by budget DSS.

**The reporting and data multidimensional analysis function**

In order to complete a multidimensional analysis of the processed and centralized data, received from the collecting, centralizing and expense allocation module, we implemented an OLAP cube using the special features offered by the Business Intelligence component of MS SQL SERVER 2005.

The OLAP cube allows the managers of an organization to visualize a series of reports having a dynamic and multidimensional structure. Therefore the expenses and income can be presented in a multidimensional structure, being assigned to cost centers, activities, year, month or quarter.

The figure 22 depicts the dimensions structure of a data analysis OLAP cube.
The income and expenses report assigned to centers, activities, accounts, years, months (quarter) are presented in figure 23.

Key Performance Indicators report function

The DSS includes a set of key performance indicators employed by the decision makers in the surveillance of income and expense evolution, enabling them to visualize graphically and in real time the budget variation.
The MS SQL SERVER 2005 environment for developing Business Intelligence applications, provide a very powerful tool in defining the key performance indicators. They are presented as graphic symbols (traffic light, arrows) which depict their status and trend.

As it is show in figure 24, together with the management of the organization where this DSS was implemented, we elaborated three such indicators which are meant to help them in the decision-making process.

![Organization’s Key Performance Indicators](image)

**Fig. 24** Organization’s Key Performance Indicators.

We consider this method as being highly efficient, helping the organization’s management to observe the status and tendency of these indicators in an easy to interpret form.
Conclusions

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

Today there are significant improvements of these decision support technologies, concerning the data storage volumes, data processing as well as data extraction, visualization and communication. The fast development of Internet and WEB technology revealed new directions in conception and development the DSS. We can conclude that the major factors having significant impact on the evolution and development of the DSS concept can be grouped as following:

- **Artificial Intelligence Technology** had a major contribution in the process of DSS functions development, improving the modeling and simulation of the decision-making process. A growing number of expert systems and intelligent agents took over the executive factor, implementing it in intelligent platforms which have the purpose to assist the decisions.

- **Client/Server Technology**, extended the DSS capability of extracting its data from the whole organization, and exposed new opportunities for implementing the DSS in a collaborative environment.

- **The Internet** sustained a considerable development in accessing and communicating the results provided by DSS. The new technologies based on WEB interfaces and portals, allowed a significant growth in the efficiency of DSS usage.

- **Database Technology**, both relational and multidimensional, led to a multidimensional and distributable layout of data, increasing DSS capacities of storing and analyzing data and information.

- **Business Intelligence Technology** induced an extension of DSS functions of storing, analyzing and forecasting. They can be brought up as examples of successful technologies derived from the hybrid support system the data warehouse, OLAP applications, and data mining techniques.

- **RAD tools and development environments**, used for conception and development the information systems, allowed the DSS functions to be developed and updated in a very short time span, offering the possibility of prototyping the system so that it could fulfill the users demands.

- **Increased volume and complexity of data and information** which the decision makers have to analyze in order to come to a decision caused an integration of diverse techniques and technologies of data filtering and data analyzing with the DSS systems.

Considering the latest developments in the DSS concept and the new technologies from the decision support area, we can conclude that the DSS tend to have a sizable meaning than just a simple application for analyzing some indicators, including in this concept all the techniques and technologies supporting the decision-making process. It can be inferred that DSS evolved from a particular form of an information system for decision support to a more general form of a concept which defines all that is called technology for the decision support. Nowadays the scientific literature, [EOMB99], [MALL00], [TURB01], [MARA03], shows that this concept covers a large spectrum of technologies as for example: executive information system, expert
systems and intelligent agents, support systems for business intelligence (data warehouse, OLAP application, data mining), support systems for group decision, and support systems for knowledge management.

The latest trends in the theoretical and practical research concerning the conception and development of DSS (including its derivatives and subsystems) bring forward more and more the usage of UML language for modeling these systems [PRAT05], [VAND06].

Regarding the conception of DSS, we consider that it must focus both on the requirements of the decision-making process and on the technological and operational context present in the organization. The technology of data storage, processing, mining and communication as well as the information system architecture has a significant influence on the DSS design and implementing process. Therefore we propose an extended employment of meta-models and UML (Unified Modeling Language) models in order to define the requirements and determine the decision-making context in such a manner that a complete scenario of a DSS coverage area will be created. Furthermore we propose the employment of prototyping techniques in the initiation and conception phase of a DSS development, which will ensure the direct communication between the user, the system and its developer. Thus we recommend the conception of a DSS to be on client/server architecture. More and more important data an information needed for the decisional process are spread over different applications and documents throughout the organization. Furthermore the decision-making factors are spread over different locations and can access the system on different time spans. Therefore the DSS has to have the capability of connecting itself on the intranet or extranet networks in order to acquire its data and to facilitate its access both from the interior and exterior of organization.

We consider that in the design and implementing phases of a DSS, besides the meta-models and UML models, CASE (Computer Aided Systems/Software Engineering) tools and RAD (Rapid Application Development) development environments should also be used, because they permit a high flexibility in updating and changing a project. Decision making requirements are very dynamic, exposing the DSS at a considerable number of changes after the conception phase. Therefore we recommend an approach of designing and implementing a DSS to be done through iterative workflow of the Unified Process, which will lead to a more stable final version of the DSS, covering to a greater extent the users’ requirements.

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 correctly 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 address most of the user’s requirements. DSS-UNIDEF creates a unified environment for conception and development of a DSS, based on the Unified Process methodology, UML, prototyping technique, and efficient development tools. Unifying all these elements through DSS-UNIDEF lead to a faster and more complete DSS, having also the possibility to create some subsystem forms which can be reused in other systems.

Concerning the delivery of an effective decision support for the budget management, we propose the employment and improvement of a DSS model based on rules and OLAP, which was extensively discussed on chapter 7 and also implemented in an organization („County Forestry Department”). The model named above, has a quite general structure which allows it to be implemented in almost all organizations who have implemented the management
by budgets. Simultaneously we want to improve and develop this model, fitting in some new functions which will allow us to determine and manage the costs through the ABC (Activity-Based Costing) method. Being designed and developed on a DSS-UNIDEF framework, updating and extending this model will prove to be very fast and highly efficient. As it was already mentioned some system models can be reused in other DSS models, therefore creating the opportunity of developing a meta-models repository.

Based on the results achieved with the research extensively described in this paper, we decided to continue it on the following directions:

- Use the new techniques, methods and unified languages (SysML) in order to determine their potential in the creation and implementation of DSS models.
- Developing the DSS-UNIDEF framework by applying it to as many DSS projects as possible.
- Development of the DSS model based on rules and OLAP by fitting in some new functions which will allow us to make use of the ABC (Activity-Based Costing) method. On this stage we will also analyze the possibility of inserting the data mining process as part of the system.
- Development of the DSS model based on rules and OLAP by fitting in some new functions which will allow us to simulate the budget and performance indicators progress.
- Analyze and exploit the practical possibility of implementing a decision-making portal as a functional integration environment for the organizational DSS.
Bibliography


*** http://en.wikipedia.org/wiki/Extreme_programming

*** http://www.extremeprogramming.org/

*** http://www.xprogramming.com/