Chapter 2
Decision Support Systems

The chapter is meant to familiarize the reader with the general notions concerning a well-defined class of information systems, namely the *decision support systems (DSS)*. The need for such systems was presented in the previous chapter, especially in Sects. 1.3 and 1.4. The chapter aims at answering several questions, such as: "In which decision situations and associated decision problems a DSS is necessary?", "What features and technology solutions characterize the class of DSSs?", "How the entities that compose the general DSS class can be classified?", "How a DSS can be constructed and what decision problems can be faced during DSS design and deployment?" The rest of this chapter is organized as follows. Section 2.1 starts with the presentation of Herbert Simon's *process model* of decision making. The characteristic features of decision making problems that make a DSS necessary are highlighted together with decision maker's limits and constraints that should be relaxed by means of a computerized support. In the same section, decision makers are classified with respect to several criteria, such as: the composition of decision units, the decision powers of participants and so on. Section 2.2 contains the general definition of DSS adopted in this book and a presentation of several technology aspects, such as: (a) the *knowledge-based framework* of Bonczek, Holsapple, and Whinston, and (b) the particular subclasses of systems and tools. A special case, namely real-time DSS for control applications is eventually described. In Sect. 2.3, we present several classifications made from various perspectives, such as: (a) number of users, (b) type of support, and (c) "dominant" technology. A brief presentation of the systems that combine numerical models with AI (Artificial Intelligence)-based technologies is made at the end of this section. In Sect. 2.4, we address the design and construction aspects as viewed from a decision-making perspective. Initially, the design and construction process is presented as an opportunity and a means to implement the change within the organization. Several design approaches are reviewed with particular emphasis on *incremental, prototype-based* one, together with the story of developing *DISPATCHER*®, a practical DSS family utilized in the continuous process industries and related fields. The general criteria for software selection are eventually

F.G. Filip et al., *Computer-Supported Collaborative Decision-Making*,
Automation, Collaboration, & E-Services 4,
DOI 10.1007/978-3-319-47221-8_2

presented together with a simple, collaborative decision making procedure based on measuring individual and group preference and polarization.

2.1 Decisions and Decision-Makers

A definition of decision was given in Sect. 1.4. One can speak about a *decision problem* when a situation that requires action shows up, there exist several possible courses of action called *alternatives*, and a *decision unit* (formed from one or several persons) is empowered to choose an alternative and is accountable for the results of decision implementation.

2.1.1 Herbert Simon's Process Model of Decision-Making

Decision-making (DM) is a specific form of information processing that aims at setting-up an action plan under specific circumstances. Nobel Prize laureate, Herbert Simon (1960) views it as a *process* made up of three steps (Fig. 2.1) as follows:

- *Intelligence*, which consists of activities such as: (a) setting the objectives, (b) data collection and analysis in order to recognize a decision problem, (c) problem statement;
- *Design*, which includes activities such as: (a) identification (or designing) possible courses of action called alternatives, (c) model building, and (d) evaluation of various potential solutions to the problem;
- C*hoice*, or selection of a feasible alternative, called decision, with a view to releasing it for implementation. Simon (1977) introduced later a fourth step which consists in *implementation* of the solution and *review* of the results.

Remark There is a subtle difference between *decision-making* and *decision-takin*g though they are commonly used interchangeably. While in the former case, the final result of the process is the chosen alternative, in the later one, the selected solution is assumed and released for implementation by an empowered person or a group of persons who are accountable for the impact of the decision taken. From now on, in the rest of the book, we will assume that it is only the human who releases the decision for implementation, or acts himself.

If a decision problem can be entirely clarified and all possible decision alternatives can be fully explored and evaluated before a choice is made, then the problem is said to be *completely structured*, otherwise is said to be *unstructured* or *semi-structured*. A structured problem occurs when the state of affairs is rather stable, the similar situations have been met in there is the past, there is no time pressure to take a highly important decision, and there are "ready made" available

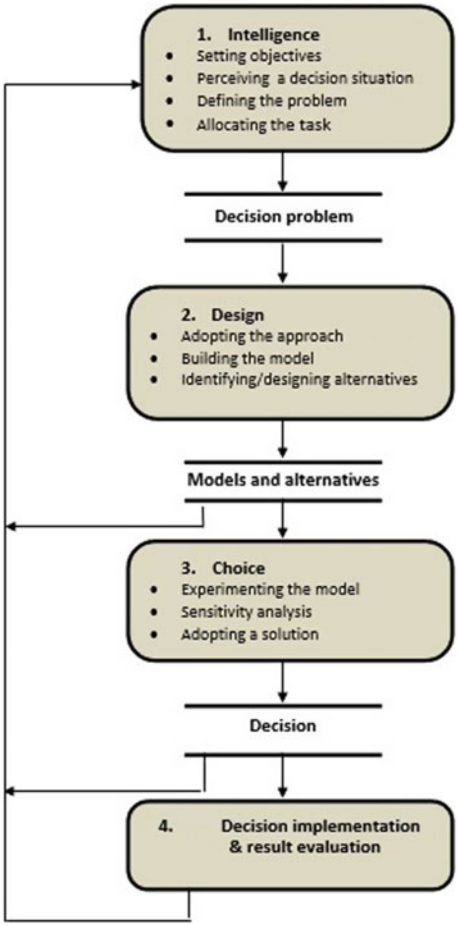

Fig. 2.1 Simon's process model of decision making

solutions. In this case, the decision is said to be *programmable*. In case of unstructured problems, no similar situations were met in the past, the available information are scarce, the consequences of a wrong decision are very serious, the time is critical. In such situations, "custom-made", *non programmable* decisions are to be made (Soelberg 1967). In Table 2.1, the attributes of completely structured and totally unstructured problems are presented.

Table 2.1 Structured and unstructured problems

	Novelty?	Information sufficiency?	Urgency?	High importance?	Programmable decisions?
Completely structured	No	Yes	No	No	Yes
Totally unstructured	Yes	No	Yes	Yes	No

If the problem is completely structured, an automation tool could provide a solution without any human intervention and, consequently, the task of selecting the course of action can be allocated to it. On the other hand, if the problem has no structure at all, only human inspiration can help. If the problem is semi-structured a computer-support for decision-making can be envisaged.

The problem structuredness level depends on several objective factors such as: (a) the characteristic features of the decision situation itself and (b) the level of decision power on which the decision unit is placed. It also may depend on subjective factors including the time limits, constraints and, even, temporary mood of the human decision-maker

2.1.2 Limits and Constrains of Human Decision Makers

The work performances of the decision-maker are influenced by several limits and constraints. Though the level of influence depends on the characteristic features of each individual decision-maker and his/her decision context, several classes of limits and constraints can be identified (Holsapple and Whinston 1996; Filip 2007) as follows:

- *Cognitive* limits concern both the quality of decision problem data available and decision procedures and techniques, methods and techniques mastered by the human decision-maker;
- *Costs* of assistants or external consultants that are possibly called to support the work of the decision-maker. Also the increased dependence of the solutions chosen on the quality of services provided by assistants may be viewed as a limit;
- *Temporal* constraints that must be observed when urgent decisions are to be made in time-critical situations or several decision problems are to be solved at the same time by the same person or group of persons;
- *Communication* or/and *collaboration* limits and constraints that show up when several persons who possess various backgrounds, knowledge bodies and intentions are involved in making a decision or/and implementing a chosen solution;
- *Low trust* of human decision-takers in the solutions recommended by computerized decision methods or/and the cost of the associated IT (software and hardware) products.

2.1.3 Classes of Decision-Makers

The term *decision unit* is a generic one. It may denote either one *individual decision-maker* (a role or a person) or an entity which is composed of several participants, sometimes called *group* or *multi-participant decision-maker* (Holsapple and

Table 2.2 Decision-maker subclasses

Attribute subclass	Number of participants?	Stable composition?	Equal powers?	Cooperating?	Human support team?
Individual	1	NA	NA	NA	NA
Unilateral	1	NA	NA	NA	Yes
Group of peers	2+	Yes	Yes	YES	Yes/No
Collectivity	2+	No	Yes	Yes/No	No/Yes
Hierarchical-organizational team	2+	Yes	No	Yes	Yes/No

Whinston 1996). Several specific subclasses of decision units are presented in Table 2.2, where *2+* means *two or more* participants, and *NA* stands for *non applicable*. The attributes that characterize subclasses are: (a) the number of members of the unit; (b) the participants' decision powers, (c) the stability (over a reasonably long period of time) of the unit composition, and (d) objectives pursuit (cooperation or competition) by participants, and (e) the existence of a *support team*. Each person that takes decisions may be supported in performing his/her activities by a team of assistants or external consultants who either are familiar with the problem domain (it is said that they possess *What-type knowledge*) or master decision methods, procedures and associated IT tools (called *How-type knowledge*). According to Holsapple and Whinston (1996) they together form a hierarchical team or a *Human Support System* (*HSS*). The typical functions of a HSS are:

- receiving and accepting decision-maker's requests for information (such as: problem data, results of an analysis, clarification or explanation of a response previously received, helps to formulate a question) or commands to acquire new information from various sources:
- issuing outputs, that can represent feedbacks to decision-maker's requests or unsolicited and proactive messages when information analysis performed indicates decision situations or undesirable behaviors of the decision-maker:
- maintaining and processing its own knowledge base.

Though the team members collaborate with a view to recommending a solution, they are not empowered and accountable for releasing it for execution. This task is to be carried-out by another person (the *unilateral* decision-maker), or a group of peers, or a hierarchically organized committee.

2.2 DSS Basic Concepts

The DSS appeared as a term in the early 70s, together with the systems meant for supporting managerial decisions. One may view that the DSS concept was anticipated by the idealized vision of Licklider (Licklider 1960) over the "precognitive" man-computer systems, which were meant to "[…] enable man and computers to co-operate on making decisions and control complex situations […]".

According to McCosh (2002), the term was first articulated by Scott-Morton (1967) in a seminar held in February 1964. As several other new concepts, DSS was initially received with great enthusiasm by a part of academia community and industry people. For example, Wagner (1981), a respected pioneer in the DSS field, hailed "the new school of thought called DSS". At the same time, the term itself and especially its usage were controversial. For example, Naylor (1982) claimed that DSS was a "redundant term meant to describe a subsystem of MIS (Management Information System)" and it was not based on a conceptual framework. Few years later, Bonczek et al. (1984), important contributors to the progress of the DSS movement, noticed that the term was "abusively used as a new label placed on various software products, in order to obtain a competitive advantage". Since then, research and development activities and many successful practical applications (Eom 2002, 2003; Kim and Eom 2004; Filip et al. 2014) have witnessed that the DSS concept definitely meets a real need and there is an ever broader application field for it as predicted Vazsonyi (1982). This development was stimulated by the change that can be noticed in the proficiency of all potential users in using the new technology. As Shim et al. (2002) noticed:

> Managers and knowledge workers in the late 1980s and 1990s are different from earlier DSS users […] and the roadblocks of the 1980s and 1990s for using IT in executive decision making are being removed. In fact, IT is now viewed as a strategic tool that is central to the pursuit of competitive advantage. Therefore, various DSS technologies will be more accepted throughout the enterprise, from operational support to executive boardrooms.

Good historical accounts on the DSS domain developments can be found in (Alter 2004; Shim et al. 2002; Power 2002b, 2008a; Hosack et al. 2012).

2.2.1 Definition and Characteristic Features

In Filip (2008), a DSS is viewed as

> An anthropocentric and evolving information system which is meant to implement the functions of a human support system that would otherwise be necessary to help the decision-maker to overcome his/her limits and constraints he/she may encounter when trying to solve complex and complicated decision problems that count.

The main characteristic features of a DSS are presented in a concise manner in Table 2.3.

Table 2.3 DSS characteristics

Name: ***Decision Support System***

Mission to relaxe the limits and constraints of the human decision-maker in making and taking a decision

Attributes

Users: decision-makers, their assistants, hi red consultants other knowledge workers

Qualities: anthropocentric, adaptive to end-user, evolving over the time

Stored knowledge

-*classes:* descriptive (about the decision problem/domain), procedural (solvers), reasoning (about combining system functions), communication (to support meessage exchanging)

- *sources*: initial design, end-user inputs, acquired from third parties or internally created

Functions: computerized version of functions of the *Human Support Systems:*

- *receiving and accepting* decision-maker's requests
- *issuing* outputs to decision –maker and requests and orders to third parties
- *maintaining and processing* its own knowledge

2.2.2 DSS Technology

2.2.2.1 Particular Subclasses

The general class of DSS can be particularized as two subclasses: (a) application oriented *specific DSS*s (SDSS) and (b) *DSS tools* (DSST). The entities of the former subclass are used by particular decision-makers (end-users) to carry-out their

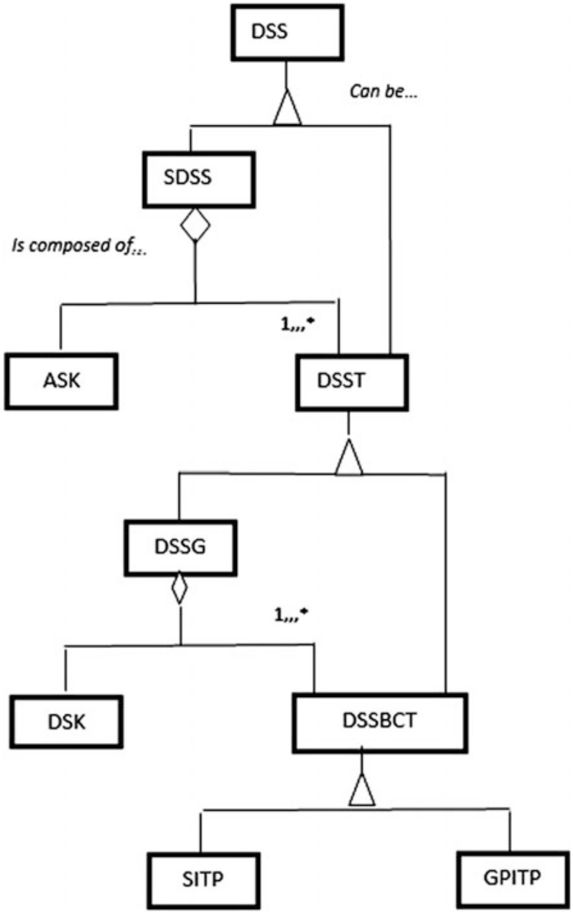

Fig. 2.2 DSS subclasses. *DSS* Decision support system; *SDSS* specific (application) DSS; *DSSBCT* DSS basic construction tool; *DSSG* DSS generator; *GDSS* group DSS; *IITT* integrated IT tool; *ITT* IT tool; *SITT* specific IT tool; *GPITP* general-purpose IT product; *GPK* General–purpose knowledge; *SK* specific knowledge; *ASK* Application-specific knowledge; *DSK* Domain specific knowledge; 1…*= "one or more"

specific tasks in a well-defined organizational and technical setting. Consequently, the systems must possess *application specific knowledge* (ASK). The objects of the later subclass are used by system builders to construct the application systems. They can be *basic construction tools* (DSSBCT) and *integrated tools* (Fig. 2.2).

The DSS basic construction tools can be *general-purpose* or *specialized IT products* (GPITP and SITP). The first category covers hardware facilities such as PCs, workstations, communication equipment, or software components such as operating systems, compilers, editors, *database management systems* (DBMS), optimization libraries, GIS (Geographical Information Systems) modules and so on. Specialized technologies are hardware and software tools such as sensors, simulators, computerized decision analysis suites, *model base management systems* (MBMS), expert system shells, other AI (Artificial Intelligence)-based software modules that have been created for building new application oriented specific DSSs or for improving the performances of the existing systems. A special subclass of tools is composed of DSS *generators* (DSSG), which are prefabricated integrated systems, oriented towards well defined application domains and functions.

The generators can be quickly adapted to particular instances of the same the application domain provided they are properly customized for the application characteristics and for the end-user's specific needs (Sprague 1980, 1987). As noticed in (Bhargava et al. 2007) "application -specific DSS are far easier to build [starting from a DSSG], but rarely reusable; DSS generators are far more complex to build, but can be adapted to build many specific systems".

An application oriented specific DSS can be developed from either a system generator (possibly using additional tools) to save time or directly from the basic construction tools to optimize its performances in particular the flexibility of the solution. While some tools can be software pieces which are bought and installed at application site, other can be hired and paid-per-use.

2.2.2.2 Architecture

Bonczek et al. (1981) proposed a DSS *generic framework*, called here *BHW model*, which is quite general and can accommodate the recent technologies and architectural variations. It is, consequently, adopted in this book and reviewed in this section. The BMW model is based on four essential components: (a) language subsystem, (b) presentation subsystem, (c) knowledge subsystem, and (d) problem processing subsystem (Holsapple 2008).

The Language Subsystem (LS) is the set of all communication means through which the end-user can transmit local or remote (via networks) messages containing problem data, requests for analysis or explanations in a form which is understandable and acceptable by the system. The LS can be also utilized by other people or automation devices that play the roles of *data feeders* or decision i*mplementers* for sending reports upon receiving requests via DSS or on their own initiatives.

The Presentation Subsystem (PS) is the set of means utilized by the DSS to send local or remote output messages (such as requests for data to be input by end-users themselves or data feeders, results of information analysis and solution selection and justification, decisions released for implementation or actions to be automatically executed) to the end-user or other people or actuators. The LS and the PS compose together the *communication subsystem* (CS).

The Knowledge Subsystem (KS) contains the pieces of *knowledge* (K) that were input into the system from various sources (system designers, users, data feeders) or were newly created within the DSS. There are several types of *primary knowledge* such as:

- *descriptive knowledge*: past, present or forecast state variables of the controlled object and its environment;
- *procedura*l knowledge (or models): optimization and simulation models;
- *reasoning* (or meta) knowledge: rules about how to use procedural knowledge together with descriptive knowledge.
- The set of *secondary knowledge* is composed of:
- *linguistic knowledge*: application vocabulary, grammar rules which are needed by the system to understand user's requests or data feeders' reports;
- *presentation knowledge* to describe the manner the information is sent to the users or to decision executers;
- *assimilative knowledge* to be utilized in adding new pieces of knowledge or discarding the obsolete knowledge items or reorganizing the existing knowledge body within the knowledge subsystem.

The Problem Processing Subsystem (PPS) contains the set of pieces of software meant to process the knowledge stored within the knowledge subsystem in accordance with the requests made by the user. The set of functions performed by the PPS include knowledge acquisition, selection, analysis, derivation, and presentation. More concretely, the PPS implements *Information analysis* and *Decision selection* functions allocated to the system (see Sect. 1.3.3) to provide answers to questions as:

- *What* happens with the controlled object? (perceiving a decision situation);
- *Which* are the causes for such a situation? (diagnosing the situation);
- *How and why t*o act? (recommending and justifying a solution);
- *What* [could happen] *if* this decision [recommended by the system or input by the human] would be implemented? (simulating the impact of the solution).

Holsapple and Whinston (1996) showed that the BHW model can accommodate, as a particular case, the largely utilized D/IDM (Dialogue/Interface, Data, and Models) paradigm (Sprague 1980; Sprague and Carlson 1982; Ariav and Ginzberg 1985) (Fig. 2.3).

2.2.3 A Special Case: Real-Time DSS for Control Applications

Though, at the beginning of the DSS movement, many people viewed the system as a tool to be used only in business applications, at present, there are numerous applications meant to support the solving of real-time control problems (Filip et al. 1983, 2002; Power 2002a, c, 2011; Van der Walle and Turoff 2008; Turoff et al. 2011). Even in the early years of the domain, there were respected voices who recognized, that the DSS class include application systems for time-critical process control settings. For example, Bosman (1987) noticed that control problems could be looked upon as a "natural extension" and as a "distinct element" of planning

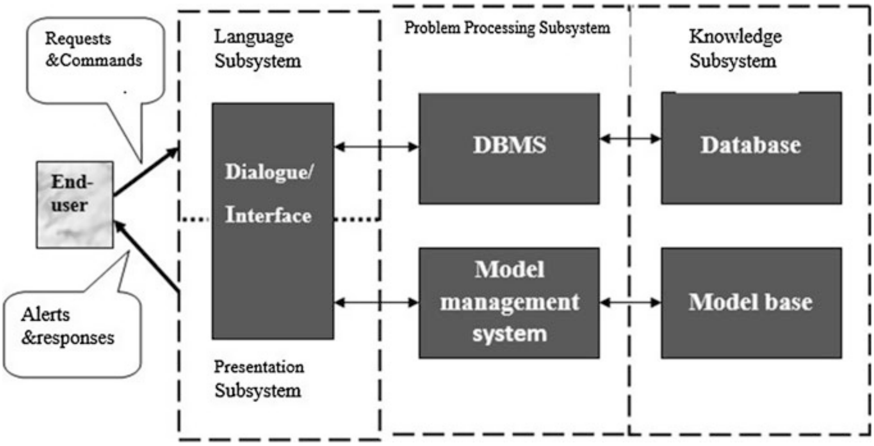

Fig. 2.3 D/IDM model as a particular case of BHW model (adapted from Holsapple and Whinston 1996)

decision-making processes. In (Sprague 1987) a DSS is meant to support communication, supervisory, monitoring, and alarming functions beside the traditional phases of the problem solving process (see Sect. 2.1.1). Chaturvedi et al. (1993) stated that real-time decision-making activities for control applications regard continuous monitoring of a dynamic environment, are short time horizon oriented, carried out on a repetitive basis, and must solve problems under time pressure.

It is unlikely that an *econological* approach (see Sect. 1.5.2.), which is based on optimization, be technically possible for a large number of genuine real-time decision-making. At the other extreme, fully automated systems (see Sect. 1.3.2), corresponding to the higher levels of automation of the classification of Sheridan and Verplank (1978) cannot be accepted either, but in some rare cases.

At the same time, one can notice that genuine real time decision-making activities can come, in most cases, across in *crisis situations*. For example, if a processing unit in a manufacturing plant must be shut down due to an unexpected equipment failure, the entire production schedule might turn obsolete. The reasonable decision is to take first adequate compensation measures to "manage the crisis" al least over the time interval which is necessary to re-compute a new production schedule or update the current one. In this case, a *satisficing* decision (see Sect. 1.5.2) may be appropriate.

In case the crisis situation has been previously faced and successfully surpassed, an almost *rule-based behavior* (see Sect. 1.3.2) based on past decisions stored in the system is an acceptable reactive approach. As a proactive measure, the minimization of the probability of occurrences of crisis situations should be considered as one of the inputs (expressed as a set of constraints or/and objectives of the model) in the scheduling problem. For example, in a pulp and paper mill or in an oil refinery plant, a stop of a processing unit may cause draining the downstream tanks which are fed by the unit plant and, at the same time, an overflow might be noticed in the upstream tank from which the unit plant is fed. Consequently, this situation implies shutting or slowing down the unit plants that are fed from up the downstream tank and so on. Re-starting up the unit normally causes transitory regimes that determine serious undesired variations in the quality of the product. To prevent such situations, the schedule made up of the sequences of production rates of processing units should be set so that stock levels in tanks could compensate to as large extent as possible for stops or significant slowing down (Filip 2008; Filip et al. 2002; Filip and Leiviskä 2009). Beside process control, another typical application domain of real-time DSS is disaster prevention and crises management (Buraga et al. 2007; Gowri et al. 2016).

From the above aspects, it results that one should add another specific necessary feature for the particular subclass of DSS to be utilized in real-time control applications (Filip 1995, 2008):

> An effective real-time DSS should support decisions on the preparation of good and cautious schedules as well as ad hoc pure RT decisions to solve crisis situations.

2.3 DSS Subclasses

As Alter (1977) remarked, in the early years of the DSS movement, "the decision support systems do not represent a homogenous category [of information systems] and one cannot speak about them in general terms". Alter compared such a wrong approach to that when "somebody speaks about pets in general without making the differences between dogs and cats or between piranha fishes and turtles". Starting from the above remark, Alter proposed taxonomy based on generic operations the system could support. The taxonomy included two main subclasses, namely *data oriented* systems and *model oriented* systems, which could be further decomposed into three and four, respectively, even more particular subclasses (Alter 1980). The taxonomy was meant to be used to facilitate the communication and to serve as a guideline for design and deployment of application oriented DSS.

Since Alter's proposal, a series of other classification schemes have been proposed based on such criteria as: user-type, level of generality (also called technology level), level of normativity, and the dominant subsystem and corresponding technology (Hackathorn and Keen 1981; Holsapple and Whinston 1996; Power 2002a). In Sect. 2.2.2, we presented a decomposition based on technology levels (Sprague 1980) of the DSS general class into two particular subclasses, namely application-oriented specific systems and DSS tools. Three additional classification schemes will be presented in the sequel, with a view to proposing a terminology to be used in the rest of the book.

2.3.1 *Classification 1 (with Respect to Decision Maker Type)*

In Sect. 2.1.3, the decision-makers were classified in accordance with a series of criteria, such as: the number of participants who take decisions, the stability of the decision unit composition and the decision position power of its members and so on. It goes without saying that a DSS must be designed to be compatible with the manner decisions are made and taken. In this respect, Hackathorn and Keen (1981) proposed structuring the general DSS class as three more particular subclasses as follows:

1. *Individual (or personal) decision support systems* (IndDSS), which are used by an individual (a role or a person) to carry-out his/her task. A special case of IndDSS is the *Executive Information/Support System* (EI/SS), which can be viewed as a natural evolution of MIS (*Management Information System*). The EI/SS can offer answers to ad hoc queries of executives and members of other staff (Watson et al. 1991).
2. *Group Decision Support Systems (GDSS)* belong to the class *group support systems (GSS)* or *electronic meeting systems* (EMS) (see Sect. 3.1). They were introduced and studied by Gallupe (1986), De Sanctis and Gallupe (1987), Gray (1987), Nunamaker et al. (1991), Gray and Nunamaker (1993), De Michelis (1986). GDSS are meant to support several individuals with similar power-positions who have to collaborate at certain time moments to make and take co-decisions (Shim et al. 2002; Gray et al. 2011; Zaraté 2013). GDSS are sometimes called *multiparticipant* DSS (Holsapple 1991; Holsapple and Whiston 1996; Marakas 2003). At present, communication capabilities, web technology and social networks play an essential role in GDSS to enable an "any-time, any-place" operation mode of the system. The following chapters will address the GDSS from various perspectives such as: concepts and methods, enabling technologies, specific features, and applications. The users of GDSS may represent the interests of various departments units of the organization. They may have different "local" priorities and possess different knowledge bodies and influence powers.
3. *Organizational DSS* (ODSS) (George 1991; Kivijärvi 1997) or *intra-organizational* DSS (Power 2001), which are meant to support those decision units composed of members that are placed on various levels of power within an organization. In case such systems are meant to support co-decisions made by people situated in different networked collaborating organizations (see Sect. 1.1), one speaks about *inter-organizational DSS* (Power 2001; Eom 2005). According to Nunamaker et al. (1991), ODSS, GDSS, and negotiation systems are specific subclasses of a more general superclass named *multiparticipant DSS*. The operation of the multiparticipant DSS may follow the models of management and control schemes described in Sect. 1.2, ranging from pure hierarchical schemes (in the ODSS case) to genuine collaborative work (in the case of GDSS).

2.3.2 Classification 2 (with Respect to Type of Support)

The next criterion of structuring the general DSS class as a set of more particular subclasses is the level of normativity of decision selection offered by the system. One can distinguish several types of support.

1. *Passive assistance* (PA): the system is supporting the end user to perform simple *data/information analysis* functions (see Sect. 1.3.2) in a comfortable and efficient manner that he/she could otherwise made manually in order to get the answers to questions, such as: "What is happening?" (see Sect. 2.2.2).
2. *Traditional support* (TS): the system mainly provides answers to questions like: "What if a certain course of action, possibly chosen by the user, would be applied?"
3. *Prescriptive Support* (PS): the system behaves as a computerized decision councilor that uses mathematical models and AI (Artificial Intelligence)-based modules to recommend a solution when the decision maker inputs his/her data into DSS. The modern *Recommender systems* (Ricci et al. 2011; Kaklauskas 2015) can be placed within the class of systems that offer prescriptive support.
4. *Collaborative support* (CS): the system enables the collaboration between the human and DSS or among various participants to decision making. In the former case, the system stimulates the user to introduce his/her solutions, then evaluates and, if necessary, refines them. It may also allow the end user to modify the solution automatically calculated (Filip et al. 1992; Filip and Leiviskä 2009). In the latter case, the system simulates the behavior of mediators (also called *facilitators* in GDSS). The modern *advisory systems* (Beemer and Gregg 2008; Kaklauskas 2015) may also be viewed as ones that offer a collaborative support.
5. *Proactive support* (PS): the system behaves as a knowledgeable and informed consultant who guides the human decision-maker in carrying-out his/her task and possibly stimulates him/her to adopt new working styles. The systems of the DIPATCHER family which will be described in Sect. 2.4 belong to PS subclass.

2.3.3 Classification 3 (with Respect to the Technological Orientation)

Alter's (1977, 1980) taxonomy was based on the extent the decision selection was influenced by the messages issued by the system. More than two decades later, Power (2001, 2002a) extended Alter's taxonomy by associating the main generic operations with so-called *dominant technologies* used. He identifies several DSS generic types as follows.

1. *Data-driven DSS* (DadDSS): the systems include, behind the old file-based technologies (*file drawers* in Alter's taxonomy), several new ones, such as:

Data Warehouse, *Online Analytical Processing* (OLAP) (Power 2008b; Bhargava et al. 2007). *Spatial DSS,* which are based on mathematical models combined with GIS (Geographical Information Systems), may be viewed in this subclass too.
2. *Model-driven DSS* (MdDSS): the systems are characterized by the predominance of numerical models that are used for evaluating through simulation the impact of possible alternatives, performing risk analysis, solving mono/multi-objective optimization problems to suggest a solution (Power and Sharda 2007).
3. *Knowledge-driven* DSS (KdDSS): which extend the meaning of *suggestion system* subclass of Alter's taxonomy. The predominant technology of KDDSS, sometimes called *intelligent DSS,* is the knowledge management (Liu et al. 2014). The subject will be addressed later in this chapter (see Sect. 2.3.4.2).
4. *Communication-driven DSS* (CdDSS): the systems are mainly meant to support exchange of data and information for collaborative/cooperative decision-making. Web-based DSS (Power and Karpathi 2002) can be placed in the CDDSS.
5. Document-driven DSS (DodDSS); such systems may be viewed as a special case of DaDDSS. In contrast with DaDDSS, which supports analysis of structured data, a DoDDSS is meant for unstructured data: web pages, multimedia documents and so on (Bhargava et al. 2007). *Business intelligence and analytics*-based systems (Hribar 2010; Power 2014; Kaklauskas 2015) fall within the subclass of DoDDSS (See Sect. 4.1).

From a historical perspective, Powers and Phillips-Wren (2011) identify seven evolution stages of DSS):

> DSS 1.0 were built using timesharing systems; DSS 2.0 were built using minicomputers; DSS 3.0 were built using personal computers and tools like Visicalc, Lotus and Excel; DSS 4.0 were built using DB2 and 4th generation languages; DSS 5.0 were built using a client/server technology on LANs; DSS 6.0 were built using large scale data warehouses with OLAP servers; DSS 7.0 were built using Web technologies.

The above authors wander whether Web 2.0 technologies have evolved enough to speak about on DSS 8.0?" The subject will be treated in Sect. 4.2.

2.3.4 Special Cases

The particular subclasses identified in accordance with the number of participant users (see Sect. 2.3.1) and dominant technology (see Sect. 2.3.3) may necessitate further nuancing. This will be made in this section.

2.3.4.1 Group Supporting Systems Versus Negotiation Support Systems

In Sect. 2.1.1, the several particular subclasses of decision units were identified in accordance with various criteria, such as the number of participants and the existence of a common goal. A distinction was made between individual and multiparticipant decision units. One can also notice that while a group was characterized by a set of relevant common objectives and a certain stability in time of its composition, the members of a *decision collectivity* may or may not share a common goal and, in addition, and they take part in a certain activity only on an episodic base.

In Sect. 2.3.1, we made a classification of DSS with respect to the numbers of participants. The subclass of group (or multiparticipant) DSS was defined for the case when there are more than one people who use the system. Final users (the decision-takers), pursue a common goal (though they might have also several local secondary objectives), or may be in an obvious competition and even in conflict. We speak, in the former case, about *group decision making* and, in the later one, about a *negotiation* meant to lead collaborative decisions that are acceptable for all parties involved in the process. Klingour and Eden (2010) state that in both cases one may speak about collaborative and interactive activities of individuals meant to reach a collective decision. As Klingour and Eden (2010) point out, "the field of group decision and negotiation exhibits both unity and diversity". The above authors accept various viewpoints about group decision and suggestion, such as:

- "The group decisions and negotiations cannot be easily detangled. There is only a minor differentiation which consists in viewing the negotiations carried out as soft social and psychological group decision activities.
- In contrast with group decision, where the decision problem is shared by more than one party who must collectively find a solution for which participants bear their specific responsibility, in group negotiation, the concerned parties may reach a collective solution or make no choice at all".

In the remaining part of the book, we will assume that the multiparticipant DSS will support the activities both in group decision and negotiation processes and highlight the differences, whenever it is necessary.

2.3.4.2 Mixed Knowledge Systems

As discussed in Sect. 1.5.2, practical field experience has shown that, in many cases, the problems are either too complex for a rigorous mathematical formulation, or too costly (with respect to time or/and money) to be solved by using available computerized numerical optimization models. As stated in that section, an optimization-based approach assumes an *econological* (economically-logical) model of the decision-making process, but, in real-life, other models of decision-making activities, such as *bounded rationality* or *implicit favorite* may be applicable to describe the actual manner the decisions are usually made and taken. Therefore, several alternative technologies based on artificial intelligence have been taken in consideration to be combined with numerical models and utilized in DSS (Dutta 1996; Dhar and Stein 1997; Filip et al. 1992; Tecuci et al. 2007). Four decades ago, Simon (1987) anticipated the combination of numerical models with the AI-based methods and tools:

> The MS/OR profession has, in a single generation, grown from birth to a lively adulthood and is playing an important role in the management of our private and public institutions. This success should raise our aspirations. We should aspire to increase the impact of MS/OR by incorporating the AI kit of tools that can be applied to ill-structured, knowledge-rich, nonquantitative decision domains that characterize the work of top management and that characterize the great policy decisions that face our society

The term *Artificial intelligenc*e (AI) is commonly used to indicate a branch of computer science aiming at making an artifact reason in a way which is similar to human manner of reasoning. Beside the classical technologies used in DSS such as computerized numerical models and data base management systems, there are several AI-based technologies (Feigenbaum 1977; Pomerol 1997; Filip and Barbat 1999; Cohen and Feigenbaum 2014; Akerkar 2014) that are usable and effectively used in DSS design

The Expert systems (ES), which were defined by E. Feigenbaum as “intelligent computer programs that use knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solution” (Akerkar 2014, p. 179). As in the case of the DSS, one can identify several particular subclasses of expert systems such as: (a) *application ES*, also called *Knowledge Based Systems* (KBS), that are systems containing problem specific knowledge, (b) system *shells*, that are prefabricated systems, valid for one or more problem types to support a straightforward knowledge acquisition and storage, (c) basic tools such as the specialized programming languages.

The similarities of the ES and DSS as the later are presented in Sect. 2.2.1 above are obvious. There are also similarities from an application-based perspective. Quite many problem types such as prediction, simulation, planning and control have been reported to be solved by using both expert systems and traditional DSS. One may view, from a DSS perspective, that expert systems are software tools to be used as components in DSS. At the same time, one from the ES side could state that DSS is merely a term denoting an application of a KBS. Even though those views might be easily explained by the backgrounds of system builders and commercial interests of IT product and service providers and consultant firms, there is indeed a fuzzy border between the two concepts, DSS and ES. However, a thorough analysis can reveal several differences between typical ESs and typical DSSs (Ford 1985). The most important ones are such as:

- the application domain is well-focused in the case of ES and is rather vague, variable, and, sometimes, even unpredictable in the case of the DSS;

- the information technology used is mainly based on symbolic computation in the ES case and is still heavily dependent on numerical models and databases in traditional DSS cases;
- the user's initiative and attitude towards the system are more creative and free in the DSs case in contrast with ES case, when the solution may be simply accepted or rejected (Filip 2005, 2008).

Case-Based Reasoning (CBR) systems' central idea consists in re-using solutions already adopted and proved to be effective in solving previously encountered problems that are similar to current decision problems. While standard expert system functioning is based on deductive processes, a CBR's one is based on induction. The operation of CBR systems includes the first or all the three following phases:

- selecting, from a case base, one (or several) decision situations which is found to be similar to the current one by using an adequate similarity measure criterion;
- adaptation of the selected case[s] to characteristic features of the problem to solve (this operation is performed by an expert system which uses to perform the reasoning on differences between the problems);
- storing and automatically indexing the just processed case for further learning and later use.

Artificial Neural Networks (ANN), also called *connectionist systems*. They are used for highly unstructured hard problems. Their operation is based on two fundamental concepts:(a) the parallel operation of several independent learning information processing units, and (b) the law which enables processors adaptation to current information environment.

As Monostori and Barschdorff (1992) remarked, expert systems and ANNs differ mainly with respect to the manner they store the knowledge. While this is a rather explicit (mainly as rules or frames) and understandable manner in the former case and implicit (as weights, thresholds) manner and hardly comprehensible by the human in case in the later one. This difference is also reflected in knowledge acquisition and modification. While knowledge acquisition is a rather complex process in case of ES and is simpler in case of ANN, the knowledge modification is relatively straightforward in case of ES but might require re-training from the very beginning in case a new element is added to ANN. An application of ANN in workshop control is given by Nicoară et al. (2011)

Agent Technology got traction since mid-1990s (Dzitac and Bărbat 2009). Franklin and Graesser (1996) define an autonomous agent as an artifact "situated within and part of an environment that senses that environment and acts on it, over time, and pursuits its own agenda and so as to effect what it senses over future". The main attributes of an intelligent agent are: (a) reactivity, (b) autonomy, (c) learning, (d) cooperation, (e) reasoning, (f) communication, and (g) mobility. It autonomously interact with its environment and makes informed decisions based on its perceptions. There are many relevant applications of agent technology in the field of medical diagnosis and prescription and multiparticipant decision units (Tecuci

1998; Bowman et al. 2001; Zamfirescu 2003; Darren et al. 2005; Tecuci et al. 2007).

Tecuci et al. (2016) developed a methodology and a corresponding set of cognitive agents that form a special subclass called *Disciple*. The agents are meant to play the role of intelligent assistants that use the evidence-based reasoning. They "learn the expertise of problem solving directly from human experts, support experts and non-experts in problem solving and decision-making and teach their expertise to students". The users, who may not be computer specialists but possess the *What-type* (application) *knowledge*, and the How-t*ype* (problem solving method) *knowledge*, teach the agents by means of examples and explanations and supervise and correct their behavior.

Young (1983) noticed that while some decision support systems are oriented towards the left hemisphere of the human brain, some others are oriented towards the right hemisphere. While in the former case, the quantitative and computational aspects prevail, in the later one, pattern recognition and the analogy-based reasoning are resorted to. Based on this observation, there is a significant trend towards combining the numerical models and the models that emulate the human reasoning to build advanced DSS.

Since the beginning of the DSS movement, numerical optimization and simulation models have been, beside datasets and database management systems, the essential characteristic constituents of DSS. Their main advantages are: compactness of data representation, computational efficiency (provided the model is correctly formulated) and the availability of a quite large number of computerized algorithms organized as high quality software libraries (see for example *NEOS Optimization Guide* to be found at: http://www.neos-guide.org/Optimization-Guide). On the other hand, numerical models may present a number of disadvantages. They are the result of intellectual processes of abstraction and idealization. Consequently the models can be applied to problems that are characterized by a certain structure. However this is hardly the case in quite many real-life problems. In addition, the use of numerical models assumes that the user possesses certain skills to formulate and experiment the numerical models (Dutta 1996).

To overcome the above problems, a solution is to view the AI-based technologies as possible complements to numerical models. The idea has been around for several decades (Turban and Watkins 1986). Various terms such as "tandem systems", "expert DSS-XDSS", "mixed knowledge DSS)" were proposed to name the systems that combine numerical models with AI based tools. A possible technology assignment of various technologies to decision making activities is given in Table 2.4 (Filip 2008).

An extended view to create *Integrated* DSS is proposed by Liu et al. (2010). The authors present a *multiperspective of integration* of the well-known components of the D/IDM framework (see Sect. 2.2.2) with modern technologies as follows:

- *Information integration* perspective: the data component is expanded to include the technologies for retrieval of data, information, and knowledge from multiple sources such as data warehouses. This can be further extended by unstructured

Table 2.4 A possible ask assignment in DSS

Activities	EU	NU	NM	ES	CBR	ANN	IA	BI&A
Intelligence								
Setting objectives	E	M			I/M			
Perception of the decision situation	I	P		P/E			P	
Problem recognition and classification	I	P			M/I	P		
Design								
Selecting the approach	E	M						
Selecting the model type	I	P						
Model building	P	P		P/I				
Model validation	I	M						
Identifying/designing alternatives	P	P	P					I
Choice								
Experimenting models								
Model solving			E			P	P	
Result interpreting	I	M		M/E				
Sensitivity analysis	I	M	I					
Decision selection	I	I			M/I		P	I
Release for implementation	E	E						

EU Expert user, *NU* Novice user, *NM* Numerical model, *ES* Rule based expert system, *ANN* Artificial neural network, *CBR* Case based reasoning, *IA* Intelligent agent, *P* Possible, *M* Moderate, *I* Intensive, *E* Essential, *BI* Business intelligence and analytics

text bases, and web content, GIS (geographical information systems) to obtain *Spatial DSS*. Corresponding processing modules for data and text analysis such as: OLAP (*On-line Analytic Processing*), data mining and text mining are, accordingly, added.

- *Model integration*: the model component is expanded to include modern business models such as: ERP (*Enterprise Resource Planning*), SCM (*Supply Chain Management*) and CRM (*Customer Relationship Management*). Qualitative models and the corresponding processing modules to provide decision suggestions and human user guiding are included.
- *Presentation* (Dialogue) *integration* by allowing combined modes and styles of man-machine communication

2.4 DSS Construction

The I&TC (*information and communication technology*) vendors continuously release to the market ever more modern hardware and software products and provide exhaustive services for project management in the information system development. In addition, new business models and technology trends are gaining traction on the market such as: increased usage of Internet, business intelligence and analytics, and viewing *I&CT/Software as a Service* in conjunction with C*loud Computing*. During the process of designing and implementing any information systems, one can face sequences of decisions which should be made, at certain time moments, with respect to the choice of the most adequate alternative concerning several critical aspects, such as: system orientation, composition of the team of the persons involved in design, method to be adopted, I&CT tools to be utilized, resources to be allocated and so on. There are various aspects, both of the technical and non-technical nature, which should be taken into consideration. Among the main aspects which might make the decisions of the designer difficult are: (a) the diversification of the possible technical constituents, and (b) the requirements for the solution quality which are set by the users who are more and more informed and have to face an ever more fierce competition. To choose the appropriate solutions for the decision situations which can be encountered in the process, an approach based *multi-attribute decision models* (MADM) could be effective.

In this section, we will use a decision-making perspective to survey several methodological and practical aspects of designing effective, *usable, useful*, and actually *utilized* DSS following and updating the aspects presented in Filip 2011, 2012; Borne et al. 2013, Chap. 7.

2.4.1 Influence Factors

There are several factors which can influence the process of designing and implementing an information system, such as: (a) the participants to the process, (b) the orientation and the purpose of the system, (c) the organizational setting, (d) standards to be utilized, and so on. They should be taken into consideration by the management team of the target enterprise and the designer as well, when a decision on creating and installing a DSS is to be made.

2.4.1.1 People

The people involved in the DSS design and implementation should cooperate closely as a team to obtain a good solution for the allocated resources (time, manpower, money). They all should take part and contribute in various extents to the process for the first moment of discussing the idea of DSS until its "steady state" operation and impact evaluation.

The following generic classes could be identified: (a)"clients", (b) designers, and (c) I&CT vendors and service providers. The members of the "client" class include the project "sponsor" (a manager), and the project "champion", who represents the interests of the future direct ("hands-on") or indirect ("beneficiary") actual *users* (who may be also participate to the design and implementation of the system). The "project champion" possesses the necessary knowledge of the application domain (the so called W*hat*-type knowledge). The "project sponsor" possesses the authority since he/she is empowered to represent the interests of the organization and, consequently, has the rights to accept or reject the project solutions and to decide on further allocating the necessary resources to continue the project. The *designers* can be members of a I&CT department of the target organization or/and a team of analysts and programmers of a consultancy firm. They master the design techniques (the *How*-type knowledge) and are familiar with the I&CT products available on the market. The I&CT *manufacturers* and *vendors,* and other service providers can adapt and alter the I&CT products and services to be utilized.

2.4.1.2 Orientation and Purpose

The information system may *be oriented* to serve a certain generic class of users ("roles") or to support a specific group of persons with names, identities and specific IT skills ("actors").

The system *purpose* might be either to facilitate and make more comfortable and efficient the work of the users, or to promote the change. In the later case, one can use the model of the *planned change* (Kolb and Frohman 1970; Sharma 2006, p. 45). The model is composed of seven steps to be performed in collaboration by the users and designers (the change agents) in the process of creating a DSS meant to facilitate implementing the change:

- *Exploring*: evaluating the needs of the organization and necessary competences of the system constructors;
- *Entry*: establishing the objectives to be agreed on by both organization empowered representatives and designers;
- *Diagnosis*: collecting the data, defining the problem and estimating the necessary resources;
- *Planning*: setting up the work plan and allocating the corresponding resources;
- *Action*: designing and implementing the DSS and train its users;
- *Stabilization and evaluation*: of the process and project impact;
- *Termination.*

2.4.1.3 Constraints and Standards

The target organization where the information system is to be implemented may strongly influence the solution and the process of the system building by imposing a set of constraints. There might be constraints caused by several factors such as: (a) the insufficient I&CT skills or/and confidence of the future users, and (b) scarce available data or/and limited internal data access rights of the external consultants. Several integration problems may show up caused by the *legacy IT systems* and infrastructure or/and the operating procedures permitted within the organization.

Nowadays, *standards* play a central role in design. The *International Standard Organization* (ISO) is an excellent source of documents to be utilized to set the stage for useful, usable and used solutions. The standards for *usable* (traditionally called "user-friendly") interfaces, such as those of the series *ISO 9241(2016)* ("Ergonomics of Human–System Interaction"), are recommended and can contribute to obtaining a user-centered solution. Other standards, such as *ISO 9241-171.2008* ("Guidance in Software Accessibility") and *ISO 9241-151.2008* ("Guidance WWW User Interface"), are of a particular importance in the context of modern information systems which are ever more oriented to use www technologies. Galvan et al. (2015) recommend the new ISO/IEC 29110 standard for Project Management Process in very small teams settings.

Other aspects, such as previous experience gained with implementing DSS within the organization, industry competitors' initiatives, legislation pressure and, the most serious ones, available budgets and intended due dates, may also influence the design and construction process.

2.4.2 Design and Implementation Approaches

There are various approaches to designing, building and implementing an information system. They can be grouped in accordance with several aspects:

- *Orientation*: software centric versus work centric;
- *Method* to be adopted: the lifecycle one or the evolving/adaptive design which is based on the use of the prototype;
- I&CT *tools and platforms* which will be utilized: general-purpose products versus integrated suites/generators/shells;
- *Source of components*: buying IT products or using *IT as a Service-ITaS*, or *Software as a Service-SaaS*;
- *Place* for construction: within the target organization or at the consultant's site.

2.4.2.1 Technology Oriented or Work Centric

In an article about *Work System Theory* (WST), Alter (2011) noticed:

> The default assumption in much of the Information Systems (IS) discipline is that systems are technical artifacts that users use, rather than sociotechnical systems in which people participate. In contrast, WST's default assumption is that human participants are essential elements of sociotechnical work systems, not just users of hardware and software. That is why the work system framework […] contains the term *participants* rather than users. A project collaboration approach would be less likely to misconstrue IS projects and IT projects. IS projects managed as work system projects might encounter less resistance and fewer surprises than IS projects managed as the creation and installation of IT artifacts.

In the particular case of collaborative systems, Briggs et al. (2009, 2015) propose a multi-layer, socio-technical design model. They remark that there are many information technology products available on the market to support collaborative activities which, however, do not necessarily assure effective collaboration. The approach recommended to obtain the expected value of the technology consists in a "combination of actors, hardware, software, knowledge, and work practices to facilitate groups in achieving their goals in effective and efficient way". Consequently, Briggs et al. (2015) identify six (initially seven) *areas of concern* for designers of collaborative systems, such as: (a) collaboration goals, (b) group deliverables, (c) group activities, (d) group procedures and techniques, (e) collaboration supporting tools, and (f) collaboration behaviours. From the general model above, one can infer that the design of a multi-participant group DSS (viewed as a collaboration supporting tool) should take place only after several prerequisite steps have been completed in sequence. The subject will be developed in Sect. 3.4 where *collaboration engineering* is described.

2.4.2.2 The Prototype-Based Approach

The traditional *lifecycle*-based method requires several steps, such as: system analysis, design, implementation, and operation which are carried out in a sequential ("cascade") manner. It also implies that the well-defined procedures and checkpoints are strictly observed and the solutions adopted are well documented. It is, consequently, recommended for large-scale applications.

The origins of the *prototype*-based method (Shelly et al. 2010) in the field of DSS design could be traced back in mid 1970s in the empiric observation that 80 % of the design ideas in the field are wrong (Ness 1975). Consequently, it was proposed, in order to avoid the waste of resources, to accept to spend 20 % of the resources in the early stages of design and construction for identifying the 80 % wrong ideas, so that the remaining 80 % of resources could be utilized to implement the remaining 20 % of ideas which are hoped to be correct.

When adopting the prototype-based method, there are a few *basic principles* which are to be observed such as:

- The process starts with approaching the most critical problems of the target organization, so that the user's confidence could be gained as early as possible. The early requirements can be formulated in collaboration with the user in a quick and even simplified manner.
- The information system is developed in several cycles which include activities such as prototype experimentation, evaluation, and modification. The cycles should be as short as possible and the cost of the first version must be very low, in order not to lose the user's interest and confidence.
- The evaluation of the usage of the preliminary versions is performed on a permanent time basis.

Two main types of prototypes have been commonly utilized (Sprague and Carlson 1982): (a) the *throwaway* prototype and (b) the *evolving* one. While the former is only utilized to test the design ideas and then is discarded (the next versions are re-designed by possibly using new technologies and methods), the latter consists in a series of improvements of the initial version.

The prototype-based (also called adaptive/incremental/iterative) methods allow for obtaining a highly customized, early utilizable and helpful solution, even though the information on organization and its business context could be, at the starting point of the process, scarce and uncertain. On the other hand, methods may cause a tendency to continually modify the solution or, on the contrary, to adopt too early a solution which is imperfect or incomplete.

In (Filip 1995; Borne et al. 2013, Chap. 7), is presented the story of constructing *DISPATCHER*®, a family of *Decision Support Systems* (DSS) which are meant to support the logistics and production control decisions to be made in the milieu of the continuous process industries and related fields. The *DISPATCHER*® project started in early 80s as an optimization model and corresponding software for production scheduling. Since then, under the influence of various factors (such as the users' changing needs and improvement of their I&CT skills, specific characteristic features of target enterprises, and new products and technologies released in the field of I&CT), several application versions were designed and deployed in industrial complexes made up of processing units interconnected through tanks, for example refineries, pulp and paper mills, chemical plants, and water systems. The initial application system has evolved towards a complex solution, a *DSS generator,* which could be adapted to new business models (such as the collaborative "extended"/"networked"/"virtual" enterprise), to support new functions and usages. It includes new constituents, such as a three-level modeling scheme of the plant (expressed in terms of final users, analysts and programmers, respectively), AI (*Artificial Intelligence*)-based guiding facilities, and model solvers and experimentation tools (Fig. 2.4).

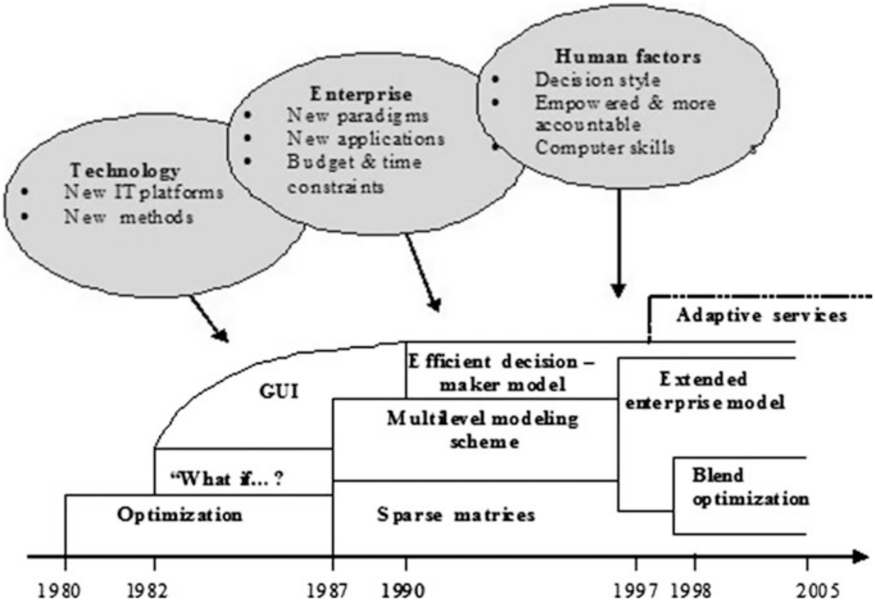

Fig. 2.4 Evolution of DISPATCHER DSS (Filip 2012; Borne et al. 2013)

2.4.2.3 I&CT Products or Services?

An important design decision consists in making a choice between *buying or leasing* I&CT products or services. For example, in recent years, the approach to use SaS (*Software as a Service*) or ITaS (*IT as a Service*) has become more and more popular. This new business model (Carraro and Chong 2006; Trumba 2007; Hine and Laliberte 2011) means that the software companies that provide SaaS services host on their servers the applications to be accessed on request, via Internet, by client organizations only when it is necessary. SaS is usually associated with cloud computing (Avram 2014).

The pricing scheme could be based on monthly lease fees. It can be also of a *pay-per-use* type instead of initial license cost and annual maintenance fees. The SaS pricing scheme is apparently of particular interest for SME (small and medium enterprises) that have limited I&CT infrastructure and do afford hiring and training their own skilled personnel. On the other hand, when a decision is to be made, one should take into account long ongoing running costs and, especially, the data security and response time constraints. In addition, the organization management should be conscious of the fact they might become a "captive client" of the service provider.

At present, evolving, modular and customizable platforms are available. An example is presented in Sect. 5.3.

2.4.3 *Selection of the I&CT Tools*

The selection of the I&CT tools should be viewed as a *multi-attribute decision-making* (MADM) problem (Filip 2005; Peng et al. 2011; Kou et al. 2012; Stefanoiu et al. 2014, Chap. 4; Zavadskas et al. 2014).

2.4.3.1 Criteria

Several general criteria to be used in selecting and ranking the possible I&CT products available on the market which are adapted from (Le Blanc and Jelassi 1989; Dhar and Stein 1997; Power 2005, p. 10, 2008b) are presented in Table 2.5.

A useful list of *pitfalls* to be avoided when selecting a software product is presented by Software Resources (2016). It includes the following 12 *deadly mistakes*: (a) buying the same software as the competitors, (b) buying software based on features alone, and overlooking other critical factors (scalability, flexibility, excessity, technology and cultural fit, affordable cost, insufficient technical support and infrastructure), (c) neglecting the proper consideration of the vendor reputation, (d) buying software without focusing on the implementation partner, (e) taking into consideration only the low initial costs and overlooking significantly

Table 2.5 Evaluation criteria

Subset	Evaluation criterion	Preferred value	Collective properties
Adequacy of methods	Accuracy of results	▲	Completeness
	Response time	▲	
	Tolerance for poor quality of data	▲	
	Explanation features	▲	
Quality of implementation	Scalability	▲	Nonredundancy
	Flexibility	▲	
	Code size	▼	
	Easy integration	▲	
	Informational and functional transparency	▲	
	Usability	▲	Decomposability
	Reliability and robustness	▲	
	Documentation completeness	▲	
Acquisition and exploitation	Delivery completeness	▲	Operability
	Price and delivery time	▼	
	Provider reputation	▲	
	Dependence on technical assistance	▼	

▲ The highest value is preferred; ▼ The lowest value is preferred higher ongoing costs, (f) buying software using input from an elite group without getting buy-in from the organization at large, (g) choosing the popular software without considering all the possible and affordable options (h) buying software that is too complex, (i) making a choice without properly defining your requirements, (j) buying software that is either at the end or at the beginning of its product lifecycle, (k) buying software that is based on a "dying technology", (l) selecting a software only to fix the current business problems, instead of implementing the change (see Sect. 2.4.1).

A systematic methodology for software evaluation and selection through the usage of MADM (Multi-attribute decision model) was proposed by Moriso and Tsoukias (1997) and an experimental expert system is described by Vlahavas et al. (1999).

At present, there exist several independent on-line services for supporting software evaluation and selection. Several examples are *Technology Evaluation Center* (TEC, 2016), Software Resources (2016), Project Perfect (PP 2016).

2.4.3.2 A Simple Collaborative Selection Algorithm

In Sect. 1.5.3.1, we presented a simple decision problem concerning the selection of the approach to be used in constructing an application information system. Section 1.5.3.3 contained several possible rules for aggregating the preferences of the people involved in the decision-making process. They took into account the decision powers of participants determined by their legal and rational position. In the sequel, a simple and practical method proposed by Coman (1996) is reviewed to give an example of a collaborative decision-making process. It is based on the *Intensity-Polarity-Voting* (*IPV*) model.

Let assume that there are:

- nd individual decision-makers, D_k (k = 1,2,…, nd);
- na possible courses of actions (alternative), A_i (i = 1,2,…, na);
- nc evaluation criteria, EC_j (j = 1,…, nc).

To aggregate individual preferences, the model uses two basic concepts: (a) the individual attitude of each member of the group towards the alternatives with respect to the set of criteria, and (b) the attitude of the group.

The individual attitude is defined by two metrics:

1. The *individual position* of the participant D_k towards the alternative A_i with respect to the evaluation criteria EC_j is expressed by a *score* (also called *attribute level*, or *perceived value*), v_{ij}^{k}, where: k, i, and j denote the participant, alternative and criterion, respectively. In addition, for each evaluation criterion, EC_j, upper and lower acceptable limits (sometimes called *thresholds*), denoted by vu_j and vl_j, respectively, should be set. In the sequel, we will assume the scores are normalized, e.g. all variation domains are defined by the same limits, for example [0.1], or [0, 10].

2. The *individual intensity* of (or the *consideration* for) the position of participant D_k with respect to evaluation criterion EC_j, is denoted by c_j^k. The intensity depends either on the power position of the participant within the decision unit or his/her perceived or/and stated determination, competence and even ability to sustain his/her own position during debates concerning the *j*-th aspect of evaluation. The intensity may take numerical values within a certain numerical interval or can be expressed through verbal statements, such as: "I believe", "It is obvious" or "I am sure on this subject matter", and "I insist". Even in the latter case, numerical values can be associated to the verbal expressions.

The group attitude is described by three metrics as follows:

1. *The group position, gv_{ij}*, towards a certain alternative, A_i, with respect to the evaluation criterion, EC_j. It is calculated as the center of gravity of individual positions:

$$gv_{ij} = v_{ij}^k c_j^k / \sum_{k=1}^{nd} c_j^k \; (i = 1, 2, \ldots, na; \; j = 1, \ldots, nc) \tag{2.1}$$

2. *Group intensity* regarding the group position towards a certain alternative, A_i, with respect to the evaluation criterion EC_j. It is denoted by gc_{ij} and is calculated as follows:

$$gc_{ij} = \sum_{k=1}^{nd} c_j^k (i = 1, 2, \ldots, na; j = 1, \ldots, nc) \tag{2.2}$$

3. The *group polarization* regarding the group collective position towards A_i with respect to EC_j measures the lack of consensus in the group about a certain aspect of the selection process. It is denoted by *gp* and is calculated as follows:

$$gp_{ij} = \sum_{k=1}^{nd} p_{ij} \, (i = 1, 2, \ldots, na; j = 1, \ldots, nc) \tag{2.3}$$

where: p_{ij} is the contribution of the participant D_k to the lack of consensus concerning the evaluation of alternative A_i with respect to evaluation criterion EC_j and is calculated as follows:

$$p_{ij} = c_j^k (v_{ij}^k - gv_{ij}) \; (k = 1, 2, \ldots, nd; \; i = 1, 2, \ldots, na; j = 1, \ldots, nc) \tag{2.4}$$

The global merit of an alternative A_i is denoted by J_i and is calculated as an aggregated utility function as follows:

$$J_i = \sum_{j=1}^{nc} w_j \, gv_{ij} \;\; (i = 1, 2, \ldots, \mathsf{na}) \tag{2.5}$$

where: the weights, $w_j(j = 1, \ldots, \mathsf{nc})$ indicate the importance of criteria. They take subunit values, are agreed in advance by all participants and fulfill the condition:

$$\sum_{j=1}^{nc} w_j = 1 \tag{2.6}$$

The group polarization on an alternative A_i is denoted by gp_i and is calculated as:

$$gp_i = \sum_{j=1}^{nc} w_j \, gp_{ij} \; (i = 1, 2, \ldots, \mathsf{na}) \tag{2.7}$$

The computation of positions, intentions and polarizations can influence the multi-participant decision processes. Thus, an alternative that possesses the highest global merit might not be chosen in case the group polarization is over a certain limit which was agreed in advance or/and the group intensity is not high enough also agreed by all participants before evaluating the alternatives. At the same time, a participant that shows an uncommonly high contribution to polarization can ask for more information and justification, with the view to possibly reevaluating his/her position. Alternatively he/she can draw the attention of the rest of the group on certain hidden aspects that can be accepted or rejected.

2.4.4 Integration and Evaluation

In some cases, a new DSS has to be integrated into the existing or planned I&CT infrastructures of the target organization. Several principles are recommended by Vernadat (1996) which are still valid for *technical integration,* such as:

- Adopting an *open system* architecture;
- *Neutralizing* the information which can be achieved by using standardized data *formats;*
- *Semantic unification* which means a symbol has a unique meaning throughout the whole system.

There are, however, several new problems which can show up due to *non-technical causes,* for example:

- *Wrong orientation* of the solution which does not facilitate solving the actual problems of the organization; this may be associated with *informational opacity* (the system provides more or less information than necessary outputs);
- *Functional opacity* which means that he user is not given the necessary information and incentives to understand how the system works;
- *Frustration* of the "hands-on" user due to a long response time or an un-adequate (insufficient or excessive) number of functions to perform his/her task.

Evaluation of information systems has been a subject of interest for both system designers and users for long time (Hamilton and Chervany 1981; Marakas 2003). There are several main principles to be observed in the process of designing, building and implementing an information system, such as:

- Evaluation is necessary in all phases of the design and implementation process. It is meant to support making a choice from the set of possible alternatives at a certain moment such as: giving up the project, continuation, supplementing or reducing the project budget, allocating additional manpower resources and so on;
- Both the set of objectives and the degree of detail of evaluation depend on various factors, as: (a) the project scope, (b) technical complexity, (c) duration and cost of the project, (d) the person who requested the evaluation, (e) overall state of the target enterprise;
- The presence of the designer in the evaluation team is necessary especially in the case of a large project.

As above stated, the evaluation is meant to support a decision-making process. Consequently, a set of *evaluation criteria* should be utilized, namely:

- Impact on of users' professional performance in accomplishing their tasks: possible additional stress caused by the DSS usage, comfort of performing the task and so on;
- The users' quality of life and their general intellectual development;
- Impact on overall evolution of the target enterprise;
- Implementation aspects and expected further running costs.

A more detailed set of criteria which was used in a specific setting is given by Al-adaileh (2009). A presentation of the methods used in multiparticipant decision-making will be made in Sects. 3.2 and 3.3.

There are several methods which can be utilized for evaluation, for example: (a) benefits/cost analysis, though the NPV ("net present value") of the investments, (b) value analysis, (c) "rating and scoring", (d) event logging and so on. Agouram (2009) proposes a useful methodology to assess the success of implementation projects and Rhee and Rao (2008) present a complete methodology to evaluate DSS. For the specific case of IDSS, Phillips-Wren et al. (2009, 2011) propose an integrative, multi-criteria design and evaluation framework based on AHP (Analytic Hierarchical Process) methodology.

2.5 Notes and Comments

This chapter closes with the enumeration of several ideas that deserve being retained by the reader as follows:

- Decision-making can be viewed as a series of information processing activities starting with setting objectives pursued and ending with releasing for implementation the chosen course of action.
- There is a subtle difference between decision-making and decision-taking. While in the former case, the result of the information processing is represented by the course of action that was chosen, in the latter, the chosen alternative is firmly adopted and then released for execution by somebody who is empowered and accountable for his/her act.
- The level of structuredness of a certain decision problem is determined by a series of attributes, such as: quality of information available in the moment the decision is made, the importance of choosing a right solution, urgency and novelty of the situation. The decision unit, the entity that makes and takes a decision, may consist in one or several persons. In the latter case, some unit members may form a human decision support team meant to facilitate the decision-making activities of the person (or persons) empowered to take the decision and release it for implementation.
- There are several cognitive, time and cost-related limits and constraints that make difficult the task of decision-making and taking.
- *The decision support system* (DSS) can be regarded as a computerized version of the human decision support team and is meant to relax the limits and constraints which could be met in solving complex decision problems that count.
- There are two main DSS conceptual frameworks: (a) the model of Bonczek, Holsapple and Whinston (composed of communication, knowledge and problem processing subsystems); (b) the D/IDM (Dialog/Interface, Data, and Model) one. The latter can be viewed as a particular form of the former.
- Real-time DSS for control represent a particular DSS subclass and are meant to support decision activities aiming at preventing and, if necessary, solving crisis situations.
- There are several DSS classification schemes according to criteria, such as: levels of generality, number of participants, type of support provided, and the dominant technology.
- There are notable tendencies to combine several technologies with a view to creating more intelligent and integrated DSS. Such systems incorporate novel and diversified technologies to support solving the ever more numerous decision problems of the present.
- The DSS design implementation process is influenced by several factors, such as: the participants, the method used, the source of software, the standards observed and so on.
- It is recommended to use a work-centric socio-technical approach, instead of a genuine software-centric process.

- Designing and implementing a DSS may represent an opportunity to manage the change implementation within the target organization.
- Prototype-based method is recommended when the uncertainties are high and the involvement of system future users in design, as co-participants, is strongly pursued.
- There are pros and cons for using the software as a service associated with cloud computing for DSS design and implementation.
- There are several groups of criteria for choosing the appropriate software pieces for a particular application-oriented specific DSS.
- In choosing the software solution, a simple and robust cooperative procedure for decision selection is exposed and recommended for utilization.

As Hosack et al. (2012), stated "DSS research is alive and well". A rather recent literature survey (Filip et al. 2014) confirms the above statement. The DSS domain is positively evolving. This is mainly due to new conceptual results, enabling information and communication technologies and ever increasing number of successful practical applications.

The chapter addressed several essential aspects of DSS domain which are valid no matter whether the decision activities are carried-out by an individual or by a multi-participant decision unit. It was, however, not possible to address al elements which deserve more detailed presentations. The interested reader can find more information in references bellow. The classical books (Alter 1980; Bonczek et al. 1981; Sprague and Carlson 1982; Holsapple and Whinston 1996; Power 2002a) and the newer ones (Turban et al. 2005; Burstein and Holsapple 2008; Power et al. 2015) can provide the reader with comprehensive presentation of the DSS domain. Particular characteristic features that are specific for the systems meant to support multi-participant collaborative decision activities will be highlighted in the following chapters. Several modern concepts in computer-mediated collaboration will be described in Chap. 3 together with the presentation of the methods used in multiparticipant settings. Major information and communication technologies that enable computer-supported multi-participant decision-making will be addressed in Chap. 4.

References

Agouram H (2009) Defining information system success in Germany. *International Journal of Information Management 29:* 129–137.

Akerkar R (2014) *Introduction to Artificial Intelligence*. PHI Learning Pvt. Ltd., Delhi.

Al-adaileh R M (2009) An evaluation of information systems success: A user perspective - the case of Jordan Telecom group. *European Journal of Scientific Research, 37* (2): 226–239.

Alter S (1977) A taxonomy of Decision Support Systems. *Sloan Management Review, 19* (1): 9–56.

Alter S (1980) *Decision Support Systems: Current Practices and Continuing Challenges.* Addison Wesley, Reading, MA.

Alter S (2004) A work system view of DSS in its fourth decade. *Decision Support Systems*, *38*(3): 319–327.

Alter S (2011) The work system method: systems thinking for business professionals. In *Proceedings of the 2012 Industrial and Systems Engineering Research Conference*, Orlando, Florida.

Ariav G, Ginzberg M J (1985) DSS design: a systemic view of decision support. *Communications of the ACM*, *28*(10): 1045–1052.

Avram M G (2014) Advantages and challenges of adopting cloud computing from an enterprise perspective. *Procedia Technology*, *12*: p. 529–534.

Beemer B A, Gregg D G (2008). Advisory systems to support decision making. In: *Handbook on Decision Support Systems 1*, Springer Berlin Heidelberg: p. 511–527.

Bhargava H K, Power D J, Sun D (2007) Progress in web-based decision support technologies. *Decision Support Systems*, *43*(4): 1083–1095.

Bonczek R H, Holsapple C W, Whinston A B (1984) Developments in decision support systems. *Advances in Computers, 23*: 141–175.

Bonczek, R.H., Holsaple, C. W., Whinston, A.B. (1981). *Foundations of Decision Support Systems.* Academic Press, New York.

Borne P, Popescu D, Filip F G, Stefanoiu D (2013) *Optimization in Engineering Sciences*. *Exact Metods*, J. Wiley& Sons, London.

Bosman A (1987) Relations between specific decision support systems. *Decision Support Systems,* 3(3): 213–224.

Bowman M, Tecuci G, Ceruti M G (2001) Application of Disciple to decision making in complex and constrained environments. In: *Systems, Man, and Cybernetics, 2001 IEEE International Conference on* (Vol. 5): p. 2932–2940).

Briggs R O, Kolfschoten G L, Vrede de G J, Albrecht C, Dean D L, Lukosch S (2009) A seven-layer model of collaboration; separation of concerns for designers of collaboration systems. In: Nunamaker Jr, J F, Currie W L eds. *Proceedings of the International Conference on Information Systems* (*ICIS'09*), paper 26.

Briggs R O, Kolfschoten G L, Vrede de G-J, Albrecht C, Lukosch S, Dean D L (2015) A six-layer model of collaboration. In: Nunamaker J F, Romano Jr N C, Briggs R O eds. *Collaborative Systems: Concept, Value, and Use*. Routledge, Taylor & Francis Group, London: 211–227.

Buraga S C, Cioca M, Cioca L (2007) Grid-based decision support system used in disaster management. *Studies in Informatics and Control-SIC, 16*(3): 283–296.

Burstein F, Holsapple C eds (2008). *Handbook on Decision Support Systems 2: Variations*. Springer Science & Business Media, Berlin.

Carraro G, Chong F (2006) *Software as a Service (SaaS): An Enterprise Perspective*, MSDN Library (available at https://msdn.microsoft.com/en-us/library/aa905332.aspx, accessed on 18. 03.2016).

Chaturverdi A R, Hutchinson G K, Nazareth D L (1993) Supporting real-time decision-making through machine learning. *Decision Support Systems, 10*(2): 213–233.

Cohen P R, Feigenbaum E A eds (2014) *The Handbook of Artificial Intelligence*. HeurisTech Press & William Heinemann Inc.

Coman A (1996) Competence, power and conflict in group decision making. *Human Systems Management*, 15 (4): 245–255.

Darren F, McGregor C, El-Masri S (2005) A survey of agent-based intelligent decision support systems to support clinical management and research. In: *Proceedings of the 2nd International Workshop on Multi-Agent Systems for Medicine, Computational Biology, and Bioinformatics*: 16–34.

De Michelis G (1996) Co-decision within cooperative processes: analysis, design and implementation issues. In: Humphrey P, Bannon L, McCosh A, Migliarese P, Pomerol J Ch eds (1996). Implementing Systems for Supporting Management Decisions: Concepts, Methods and Experiences. Chapman & Hall, London.: p. 124–138.

DeSanctis G, Gallupe R P (1987) A foundation for the study of group decision support systems. M*anagement Science, 33* (5): 589–609.

Dhar V, Stein R (1997) *Intelligent Decision Support Methods: The Science of Knowledge Work*. Prentice Hall, Upper Saddle River, New Jersey.

Dutta A (1996) Integrated AI and optimization for decision support: a survey. *Decision Support Systems, 18*(3): 213–226.

Dzitac I, Bărbat B E (2009) Artificial intelligence + distributed systems = agents. *International Journal of Computers Communications & Control*, 4(1): 17–26.

Eom S B (2002) *Decision Support Systems Research (1970–1999) as Cumulative Tradition and Reference Disciplines*, Edwin Mellen Press, Lewiston, NY.

Eom S B (2003*) Decision Support Systems Research and Reference Disciplines (1970–2001): A Research Guide to the Literature and an Unobtrusive Bibliography with Citation Frequency,* Edwin Mellen Press, Lewiston, NY.

Eom S B (2005) Evolution of DSS. From single user DSS to inter-organizational DSS. In: Eom S B editor, *Inter-organizational Information Systems in the Internet Age* Idea Group Publishing, London, 231–247.

Feigenbaum E A (1977) *The Art of Artificial Intelligence. 1. Themes and Case Studies of Knowledge Engineering* (No. STAN-CS-77–621). Stanford Univ CA Dept of Computer Science.

Filip F G (1995) Towards more humanized real-time decision support systems. In: Camarinha-Matos L, Afsarmanesh H eds, *Balanced Automation Systems: Architectures and Design Methods*, Chapman & Hall, London: p. 230–240.

Filip F G (2005) *Decizie asistata de calculator; decizii, decidenti, metode de baza si instrumente informatice associate* ("Computer–Aided Decision-Making; Decisions, Decision-Makers. Basic Methods and Software Tools"), 2nd Edition, Editura Tehnica, Bucuresti (in Romanian).

Filip F G (2007) *Sisteme support pentru decizii* ("Decision Support Systems"), 2nd Edition, Editura Tehnica, Bucuresti (in Romanian).

Filip F G (2008) Decision support and control for large-scale complex systems. *Annual Reviews in Control*, *32*(1): 61–70.

Filip F G (2011) Designing and building modern information systems: a series of decisions to be made. *Computer Science Journal of Moldova* 19(2): 119–129.

Filip F G (2012) A decision-making perspective for designing and building information systems. *International Journal of Computers Communications & Control* 7(2): 264–272.

Filip F G, Barbat B (1999) *Informatica industriala. Noi paradigme si aplicatii* ("*Industrial informatics*. *New Paradigms and Applications").* Editura TEHNICA (Technical Publishers), Bucharest (in Romanian].

Filip F G, Donciulescu D A, Filip Cr I (2002) Towards intelligent real time decision support systems. *Studies in Informatics and Control-SIC, 11*(4): 303–312.

Filip F G, Donciulescu D A, Neagu G (1983) Job scheduling optimization in real-time production control. *Computers in Industry, 4*(4): 395–403.

Filip F G, Leiviskä K (2009) Large-scale complex systems. In: *Springer Handbook of Automation*, Nof SY editor. Springer, Berlin Heidelberg: p. 619–638.

Filip F G, Roberts P D, Zhang J (1992) Combined numeric – knowledge based hierarchical control; part II: process scheduling and coordination. *Studies in Informatics and Control – SIC 1 (4*): 267–283.

Filip F G, Suduc A M, Bîzoi M (2014) DSS in numbers. *Technological and Economic Development of Economy*, *20*(1): 154–164.

Ford F N (1985) Decision support systems and expert systems: a comparison. *Information & Management*, *8*(1): 21–26.

Franklin S, Graesser A (1996) Is it an agent, or just a program?: A taxonomy for autonomous agents. In: *Intelligent agents III agent theories, architectures, and languages*. Springer Berlin Heidelberg: p 21–35.

Gallupe R B (1986). Experimental research into group decision support systems: practical issues and problems. In: *Proceedings of the 19th Hawaii International Conference on Systems Science*. Honolulu, Hawaii, January: p. 513–523.

Galvan S, Mora M, O'Connor R V., Acosta F, Alvarez F (2015). A Compliance Analysis of Agile Methodologies with the ISO/IEC 29110 Project Management Process. *Procedia Computer Science*, *64*:188–195.

Gowri S, Vigneshwari S, Sathiyavathi R, Lakshmi T K (2016) A framework for group decision support system using cloud database for broadcasting earthquake occurrences. In: *Proceedings of the International Congress on Information and Communication Technology*, Springer: p. 611–615.

George J F (1991) The conceptualization and development of organizational decision support systems. *Journal of Management Information Systems, 8* (3): 109–125.

Gray P (1987) Group decision support systems. *Decision Support Systems*, *3*(3): 233–242.

Gray P, Nunamaker J F (1993) Group decision support systems. In: Decision Support Systems: Putting Theory into Practice. Third Edition. Prentice Hall International Inc., London.: p. 309–326.

Gray P, Johansen B, Nunamaker J, Rodman J, Wagner G R (2011) GDSS past, present, and future. In: *Decision Support: An Examination of the DSS Discipline*. Schuff D, Paradice D, Burstein F, Power D J, Sharda R eds. Springer New York: p. 1–24.

Hackathorn, R D, Keen P G W (1981) Organizational strategies for personal computing in decision support systems. *MIS Quarterly, 5*(3), 21–27.

Holsapple C W (1991) Decision support in multiparticipant decision makers. *Journal of Computer Information Systems*, *31*(4): 37–45.

Hosack B, Hall D, Paradice D, Courtney J F (2012) A look toward the future: decision support systems research is alive and well. *Journal of the Association for Information Systems*, *13*(5): 315.

Hamilton S, Chervany N L (1981) Evaluating information system effectiveness - Part I: Comparing evaluation approaches. *MIS Quarterly, 5* (3): 55–69.

Hine J, Laliberte B (2011) *Enabling IT-as-a-Service*. White Paper. ESG (available at: http://docs.media.bitpipe.com/io_10x/io_102870/item_483065/enabling%20it-as-a-service.pdf, accessed on 18.03.2016).

Holsapple C W (2008) DSS architecture and types. In: *Handbook on Decision Support Systems 1*, Burstein F, Holsapple C eds. Springer, Berlin Heidelberg: p. 163–189.

Holsapple C W, Whinston A B (1996) *Decision Support Systems: A Knowledge-based Approach.* West Publishing Co., Minneapolis/St. Paul, Minnesota.

Hosack B, Hall D, Paradice D, Courtney J F (2012) A look toward the future: Decision Support Systems research is alive and well. *Journal of the Association for Information Systems*, *13*(5): 315–340.

Hribar Rajterič I (2010) Overview of business intelligence maturity models. Management: *Journal of Contemporary Management Issues, 15*(1): 47–67.

ISO 9241 (2016). *Ergonomics of human-system interaction.* (available at: http://www.iso.org/iso/search.htm?qt=9241&sort=rel&type=simple&published=on; accessed on 18.03.2016).

Kaklauskas A (2015) *Biometric and Intelligent Decision Making Support*. Springer Verlag, Berlin.

Kim E, Eom S B (2004) *Decision Support Systems Applications Research: A Bibliography (1995–2001)*, Working Paper, Southern Missouri State University.

Kivijärvi H (1997) A substance-theory-oriented approach to the implementation of organizational DSS. *Decision Support Systems*, *20*(3): 215–241.

Klingour M, Eden C (2010) Introduction to the handbook of group decision and negociation. In: *Handbook of Group Decision and Negotiation,* Klingour M, Eden C, eds, Springer Science + Business Models, Dordrecht: p. 1–7.

Kolb D A, Frohman A L (1970) An organization development approach to consulting. *Sloan Management Review, 12*(1): 51–65.

Kou G, Lu Y, Yi P, Shi Y (2012) Evaluation of classification algorithms using CDM and Rank Correlation, *International Journal of Information Technology & Decision Making, 11*(1),:197–225.

Le Blanc L A, Jelassi MT (1989) DSS software selection: a multiple criteria decision methodology. *Information & Management*, *17*(1): 49–65.

Licklider J C R (1960). Man–computer symbiosis. *IRE Transactions on Human Factors in Electronics,* Vol. HFE1, 4–10.

Liu S, Duffy A H, Whitfield, R I,, Boyle I M (2010) Integration of decision support systems to improve decision support performance. *Knowledge and Information Systems*, *22*(3): 261–286.

Liu S, Smith M H, Tuck S, Pan J, Alkuraiji A, Jayawickrama U (2014) Where can knowledge-based decision support systems go in contemporary business management—A new architecture for the future. *Decision Support Systems (KB-DSS)*, *1*(5): 499–504.

Marakas G M (2003) *Decision Support Systems in the 21st Century*. Prentice Hall, Upper Saddle River, New Jersey.

McCosh A M (2002) E-mail to D. Power on 03 oct. 2002. In: (Power, D., 2002 b).

Monostori L, Barschdorff D. (1992) Artificial neural networks in intelligent manufacturing. *Robotics and Computer-Integrated Manufacturing*, *9*(6): 421–437.

Moriso M, Tsoukias A (1997) JusWare: a methodology for evaluation and selection of software products. *IEE Proc. Softw. Eng. 144* (2): 162–174.

Naylor T H (1982) Decision support systems or whatever happened to M.I.S?, *Interfaces, 12* (4): 92–94.

Nicoară E S, Filip, F.G, Paraschiv N (2011) Simulation-based optimization using genetic algorithms for multi-objective flexible JSSP. *Studies in Informatics and Control*, *20*(4): 333–344.

Nunamaker J F, Dennis A R, Valacich J S, Vogel D, George JF (1991) Electronic meeting systems. *Communications of the ACM*, *34*(7): 40–61.

Ness D N (1975) Decision support systems. Theory and design. *Wharton Office of Naval Research on DSS*. Philadelphia, November: p. 4–7.

Peng Y, Kou G., Wang G, Shi Y (2011) FAMCDM: A fusion approach of MCDM methods to rank multiclass classification algorithms. *Omega*, *39*(6): 677–689.

Phillips-Wren G, Mora M, Forgionne G (2011) Assessing Today: Determining the Decision Value of Decision Support Systems. In *Decision Support*. Springer New York: 203–220.

Phillips-Wren G, Mora M, Forgionne G A, Gupta J N (2009) An integrative evaluation framework for intelligent decision support systems. *European Journal of Operational Research*, *195*(3): 642–652.

Pomerol J C (1997) Artificial intelligence and human decision making. *European Journal of Operational Research*, *99*(1): 3–25.

Power D J (2001) Supporting decision-makers: an expanded framework. In *e-Proceedings Informing Science Conference, Challenges to Informing Clients: A Transdisciplinary Approach, Krakow, Poland*: p. 431–436.

Power D J (2002a) *Decision Support Systems: Concepts and Resources for Managers*. Quorum Press. Westport, Connecticut.

Power D J (2002b) A brief history of decision support systems. *DSS Resources. COM, World Wide Web* (version 4.0 2007 available at: http://DSSResources.COM/history/dsshistory.html, consulted on: 12.03.2016).

Power D J (2002c) When is “real – time” decision support desirable and needed? *DSS News, 3* (25), 08 December.

Power D J (2005) *Decision Support Systems. Frequently Asked Questions.* iUniverse Inc., New York.

Power D J (2008a) Decision support systems: a historical overview. In *Handbook on Decision Support Systems,* Springer Berlin Heidelberg: p. 121–140.

Power D J (2008b) Understanding data-driven decision support systems. *Information Systems Management, 25*(2):149–154.

Power D J (2011) Challenges of real-time decision support. In: *Supporting Real Time Decision-Making*, Annals of Information Systems 13, Burstein F et al eds. Springer Science + Business Media, LLC: p. 3–11.

Power D J, Phillips-Wren G (2011) Impact of social media and Web 2.0 on decision-making. *Journal of decision systems*: *20*(3), 249–261.

Power D J (2014) Using ‘Big Data’for analytics and decision support. *Journal of Decision Systems*, *23*(2): 222–228.

Power D J, Karpathi S (2002) Building web-based decision support systems. *Studies in Informatics and Control.-SIC, 11*(4): 291–302.

Power D J, Sharda R (2007) Model-driven decision support systems: concepts and research

Power D J, Sharda R, Burstein F (2015) *Decision Support Systems*. John Wiley & Sons, Ltd.

PP (2016). *Project Perfect: Project Management Software*. (http://www.projectperfect.com.au/, accessed on 18.03.2016).

Rhee C, Rao H R (2008) Evaluation of decision support systems. In: Burstein F, Holsapple C eds. *Handbook on Decision Support Systems 2: Variations.* Springer Berlin Heidelberg: p. 313–327.

Ricci F, Rokach L, Shapira B, Kantor P (2011). *Recommender Systems Handbook*. Springer Science + Business Media, LLC.

Scott Morton M S (1967) *Computer – Driven Visual Display Devices: Their Impact on the Management Decision-Making Process*. Doctoral Dissertation, Harvard Business School.

Sharma R S (2006) *Change Management.* Tata Mc Graw- Hill, New Delhi.

Shelly G B, CashmananT J, Rosenblatt HJ (2010) *Systems Analysis and Design*, 8th Ed., Thomson Course Technology, Boston, Mass.

Sheridan T B, Verplank W (1978) *Human and Computer Control of Undersea Teleoperators.* Man-Machine Systems Laboratory, Dept. of Mechanical Engineering, MIT, Cambridge, MA.

Shim J P, Warkentin, M, Courtney J F, Power D J, Sharda R, Carlsson C (2002) Past, present, and future of decision support technology. *Decision support systems*, *33*(2): 111–126.

Simon H. (1960) *The New Science of Management Decisions*. Harper Row, NewYork.

Simon H. (1977) *The New Science of Management Decisions*. Revised Edition. Prentice Hall, Englewood Cliffs, New Jersey.

Simon H (1987) Two heads are better than one: the collaboration between AI and OR. *Interfaces*, *17*(4): 8–15.

Soelberg P O (1967). Unprogrammed Decision Making. *Industrial Management Review*, 11 pg.

Sprague Jr, R H (1980) A framework for the development of decision support systems. *MIS Quarterly*, *4(*4), 1–26.

Sprague Jr R H (1987) DSS in context. *Decision Support Systems, 3,* 197–212, republished in: (Sprague, Watson eds, 1993): p. 29–40.

Sprague Jr., R H, Carlson E D (1982) *Building Effective Decision Support Systems*. Prentice Hall, Englewood Cliffs, N. J.

Software Resources (2016) *12 Deadly Mistakes of Software Selection* (available at: http://www.softresources.com/resources/articles/12-deadly-mistakes-of-software-selection, accessed on 18. 03.2016).

Software Resources (2016) *Smart Software Selection* (available at: http://www.softresources.com, accessed on 18.03.2016)

Stefanoiu D, Borne P, Popescu D, Filip F G, El Kamel A (2014) *Optimization in Engineering Sciences: Metaheuristic, Approximate Methods and Decision Support*. John Wiley & Sons, London.

TEC (2016) *Technology Evaluation Center.* (http://www.technologyevaluation.com/ accessed on 18.03.2016).

Tecuci G (1998) *Building Intelligent Agents: An Apprenticeship Multistrategy Learning Theory, Methodology, Tool and Case Studies*. Academic Press, San Diego, CA.

Tecuci G., Marcu D, Boicu M, Vu L (2007) Mixed-initiative assumption- based reasoning for complex decision-making. *Studies in Informatics and Control-SIC, 16*(4), 459–468.

Tecuci G, Marcu D, Boicu M, Schum D A (2016) *Knowledge Engineering: Building Cognitive Assitants for Evidence-based Reasoning*. Cambridge University Press.

Trumba (2007). *White Paper: Five Benefits of Software as a Service* (available at: http://www.trumba.com/connect/knowledgecenter/software_as_a_service.aspx, accessed on 09.02.2012).

Turban E, Aronson J E, Liang TP (2005) *Decision Support Systems.* 7th ed., Pearson – Prentice Hall, Upper Saddle River, New Jersey.

Turban E, Watkins P R (1986) Integrating expert systems and decision support systems. *MIS Quarterly*, 10(2): 121–136.

Turoff M, White C, Plotnick L (2011). Dynamic emergency response management for large scale decision making in extreme hazardous events. In: *Supporting Real Time Decision-Making*, Annals of Information Systems 13, Burstein F et al eds. Springer Science + Business Media, LLC:: p. 181–202.

Van de Walle B, Turoff M (2008) Decision support for emergency situations. *Information Systems and E-Business Management*, *6*(3): 295–316.

Vazsonyi A (1982) Decision support systems, computer literacy and electronic models. *Interfaces, 12* (1): 74–78.

Vernadat A (1996) *Enterprise Modeling and Integration Principles and Applications*. Chapmann & Hall, London.

Vlahavas I, Stamelos I, Refanidis I, Tsoukias A (1999) ESSE: an expert system for software evaluation. *Knowledge-Based Systems, 12*(4): 183–197.

Wagner G R (1981) Decision support systems: The real substance. *Interfaces*, *11*(2): 77–86.

Watson H J, Rainer Jr R K, Koh C E(1991) Executive information systems: a framework for development and a survey of current practices. *MIS quarterly* 15(1*)*: 13–30.\

Young L F(1983) Computer support for creative decision-making: right-brained DSS. *Processes and Tools for Decision Support*: p. 2001–2004.

Zamfirescu C B (2003) An agent-oriented approach for supporting self facilitation for group decisions. *Studies in Informatics and Control-SIC, 12*(2): 137–148.

Zaraté P (2013) *Tools for Collaborative Decision-making*. John Wiley & Sons.

Zavadskas E K, Turskis Z, Kildienė S (2014) State of art surveys of overviews on MCDM/MADM methods. *Technological and Economic Development of Economy*, *20*(1): 165–179.

http://www.springer.com/978-3-319-47219-5

Computer-Supported Collaborative Decision-Making
Filip, F.G.; Zamfirescu, B.-C.; Ciurea, C.
2017, XVI, 216 p. 34 illus., Hardcover
ISBN: 978-3-319-47219-5