Management Information Systems

Book · January 2014

5 authors, including:

Mariusz Grabowski
Cracow University of Economics

Piotr Soja
Cracow University of Economics

Ryszard Tadeusiewicz
AGH University of Science and Technology in Kraków

Jan Trąbka
Cracow University of Economics

Management Information Systems

Management Information Systems

Mariusz Grabowski, Piotr Soja, Ryszard Tadeusiewicz,
Jan Trąbka, Agnieszka Zając

CRACOW UNIVERSITY OF ECONOMICS

EDUCATION FOR ENTREPRENEURSHIP

Reviewer (Recenzent):
dr hab. inż. Krzysztof Boryczko, prof. AGH

Authors (Zespół autorów):
Mariusz Grabowski, Piotr Soja, Ryszard Tadeusiewicz, Jan Trąbka, Agnieszka Zając

The Project Leader (Lider projektu):
Cracow University of Economics (Uniwersytet Ekonomiczny w Krakowie)
ul. Rakowicka 27, 31-510 Kraków
tel.:+48 12 293 57 00 lub +48 12 293 52 00
faks: +48 12 293 50 10
www.uek.krakow.pl

Project Partner (Partner projektu):
Association 'Education for Entrepreneurship' (Stowarzyszenie „Edukacja dla Przedsiębiorczości")
ul. Orląt Lwowskich 2/2, 31-518 Kraków
tel.:+48 12 430 18 11
office@edp.org.pl
www.edp.org.pl

ISBN 978-83-64509-09-4

Free of charge (Publikacja bezpłatna)

Coverage, layout, typesetting on behalf of the Cracow University of Economics (Opracowanie wydawnicze, graficzne, skład i łamanie na zlecenie Uniwersytetu Ekonomicznego w Krakowie)
Grupa Wydawnicza LUMINA Sp. z o.o.

Correction of content-language translations into English (Korekta merytoryczno-językowa tłumaczenia na język angielski):
Michał Jezierski, Lingua Lab, www.lingualab.pl

The publication elaborated within the project "Launching a unique faculty Applied Informatics in response to labour market demand" is co-founded by European Union in the frame of European Social Fund.
(Wydawnictwo przygotowane w ramach projektu „Uruchomienie unikatowego kierunku studiów Informatyka Stosowana odpowiedzią na zapotrzebowanie rynku pracy" jest współfinansowane ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.)

TABLE OF CONTENTS

INTRODUCTION

This textbook has been developed for the unique field of study of *Applied Computer Science*, co-financed by the European Union under the *European Social Fund*, launched at the Cracow University of Economics in 2010. The textbook structure was largely determined by the need to include curriculum content discussed in the bachelor's degree courses, especially in *Introduction to Information Systems*, *Analysis and Design of Information Systems* and *Enterprise Systems*.

The book consists of three chapters.

The first chapter presents an overview of the role and place of an information system within a business organisation and identifies the nature of the MIS, as an area lying at the intersection of several fields: management science, computer science, quantitative methods, operations research, sociology, psychology and economics. The first section discusses the relationship between data, information, knowledge and wisdom, which is an introduction to the essence of information processing in an organisation. These concepts are then elaborated on through a detailed review of the definitions of information found in the literature, divided into the quantitative and the qualitative approaches, and with the presentation of main theories that represent them - the quantitative theory of information by C. E. Shanonn and the infological theory of information by B. Langefors. Next, the definitions of an information system are reviewed, identifying the importance of information for organisations and discussing its attributes. The role played by an information system in the management process is explained, presenting the typology of computer-aided information systems. Also, the chapter explains the importance of information technology as a factor increasing the competitiveness of modern organisations. At the end, the concept of *IT governance* is described, which is a new approach to managing the IT area, having the features of a systemic approach.

The second chapter is devoted to the issues of the analysis and design of information systems. Its introduction focuses on the soft system approach, which is a methodological paradigm in social systems. The chapter then discusses issues related to the project life cycle and the system life cycle, thus indicating the complexity of the work associated with the process of creation and implementation of information systems. The next part of the second chapter contains an overview of basic methodologies, methods and tools, divided into the structural and the object-oriented approach. For each of the approaches, the use of appropriate tools is presented using the example of a specific system design. The chapter also discusses computer-aided design in the form of CASE tools. The next section discusses selected aspects of project management, with particular reference to PRINCE2 and SCRUM methodologies and concepts related to change management. The last part of the chapter presents the results of the research work carried out by one of the authors of this textbook related to the design of UBMSS systems, which combine computer systems with cognitive science.

The third chapter provides an overview of basic issues related to one of the most important classes of information systems today, i.e., enterprise systems. The chapter starts with a discussion of matters related to the process orientation of the organisation, essential to the efficient and effective implementation of systems of this type. Next, the chapter features an overview of definitions, a discussion on the key features and architecture, and a presentation of selected modules of an enterprise system using SAP R/3 as an example. The next section discusses the evolution of enterprise systems, from inventory control systems to ERP II systems. A large part of the chapter is devoted to motivation and successful implementation, the benefits of using an enterprise system and the determinants of the implementation process. At the end, the chapter presents the evaluation of the application of an enterprise system within an organization.

An extensive list of references used to write this textbook is included at the end. It can be used to extend the knowledge on the matters discussed, which the authors encourage the readers to do.

1

THE PLACE AND ROLE OF AN IS IN BUSINESSES

1.1. INTRODUCTION

The enormous changes in the use of information technology in social and economic life, taking place in the last five decades, are accompanied by the development of *information systems* (IS), which sometimes, because of the organisational and economic context of the discussion, are interchangeably referred to as *management information systems* (MIS) (Laudon and Laudon, 2002; McNurlin and Sprague Jr., 2002, Checkland and Holwell, 1998, Rainer and Cegielski, 2011)[1].

Management information systems are a domain of applied science aiming to define the theoretical basis needed to solve practical problems arising from the multidimensionality of these changes. Research activity is being developed as a part of MIS, and its related topic are an integral part of undergraduate and graduate level curricula (Kuraś et al., 1999; Ives et al., 2002).

The main focus of MIS, according to K. Lyytinen and J.L. King (2004, p. 221), is:

> *market of ideas in which scholars (and practitioners) exchange their views regarding the design and management of information and associated technologies in an organized human enterprise.*

Management information systems are an interdisciplinary field, whose essential elements are the systems approach and the concept of information. Systems approach defines the methodological aspect of the field. Generally speaking, it means the adoption of a set of concepts enabling understanding and describing the complexity of the real world. MIS focuses on information and, in particular, on its role in the system. The third of the terms included in the name of the field defines the context of application of information systems. This context is an organisation, usually of a commercial nature.

Management information systems stem from many different fields, disciplines and areas of science and practice. They primarily include: economics, sociology, psychology, organizational science, computer science and quantitative methods, in particular operations research (Laudon and Laudon, 2002, p. 14). Individual researchers and practitioners differently

---

[1] The term *management information systems* has begun to be supplanted by the term *information systems in the English-language literature since the late 80.* These terms are used interchangeably in this textbook.

accentuate these components. Therefore, we can talk about the hard, or technical, approach, which is dominated by IT and operations research, and soft, or behavioural, approach – dominated by sociology and psychology. Methodologically, the hard appraoch is represented by a general systems theory (Bertalanffy, 1968), while the link connecting the soft approach is the soft systems methodology (Checkland, 1993, Checkland and Holwell, 1998). Relationships between different approaches are illustrated in Fig. 1.1.

This textbook, similarly to (Laudon and Laudon, 2002), uses a holistic, or socio-technical perspective on MIS. It is a compromise between the soft approach issues i.e., economics, psychology and sociology (marked light grey in Fig. 1.1), and the hard approach i.e., IT, organisation and management, and operations research (marked dark grey in Fig. 1.1). In our opinion, only such a balanced perspective can capture the complexity of relationships and mutual conditions occurring in MIS and thus enable appropriate problem solving related to the efficient and effective implementation of IT solutions in practice.

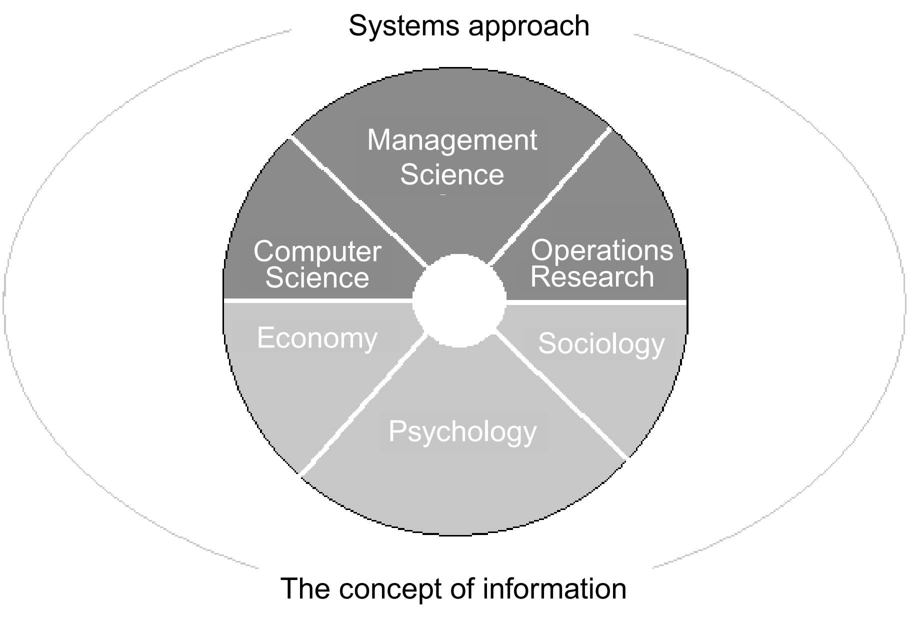

Fig. 1.1 Management information systems as an interdisciplinary field
Source: Adapted from (Laudon and Laudon, 2002, p. 14)

The Polish monographs on MIT primarily include the works under the editorship of E. Niedzielska (1997), A. Nowicki (1997, 2006), A. Rokicka-Bronitowska (2004), S. Wrycza (2000, 2010), J. Zawiła-Niedzwiecki *et al.* (2010), and independent work of W. Flakiewicz (2002), J. Kisielnicki and H. Sroka (2005) and J. Kisielnicki (2008).

1.2. BASIC CONCEPTS: DATA, INFORMATION, KNOWLEDGE, SYSTEM

Data are a basic material occurring in information systems. Each object in the real world, every process and every phenomenon can be a source of data. Data elements consist of various *facts* about these objects of the real world, recorded manually (by humans) or automatically (by the various sensors and transducers). Data are the raw records of these facts, which are not yet assigned any particular meaning or purpose. Data may take the form of numbers (usually), text, logical decisions (we determine that something is true – or not), as well as signals from different recorders, recorded sounds, images or video. Data generally are abundant and they constitute the basis (foundation) of the *information pyramid* (Fig. 1.2). However, their value is very limited because they are not arranged in order.

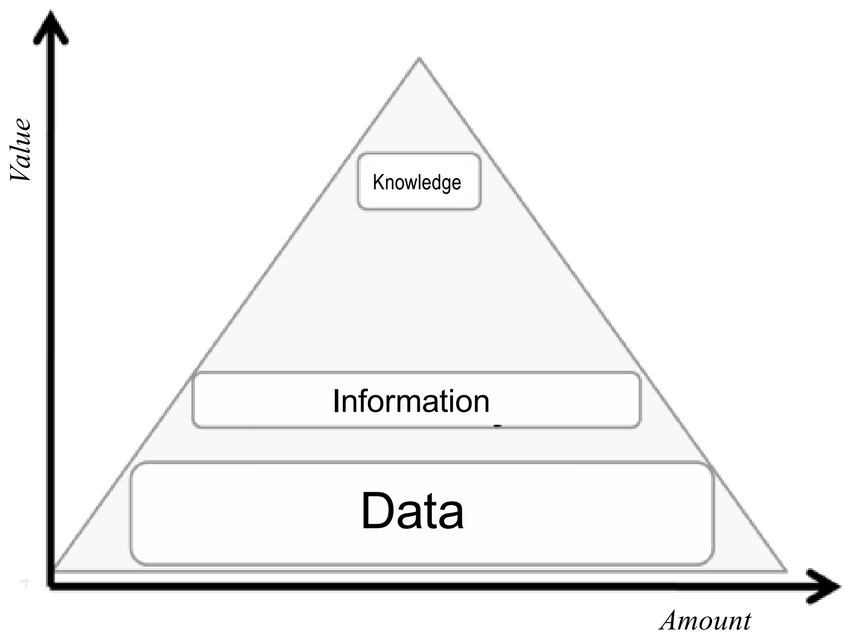

Fig. 1.2 Information pyramid
Source: Own work

Data processing results in information. There is less information than there are data, but it is of greater value, since it is possible to assess the current situation and make decisions on its basis – for example, by inference.

The transition from data to information usually consists of grouping and structuring of data and the calculation of indicators (e.g. statistical indicators) that characterise the data, but are less susceptible to random fluctuations occurring in the described reality. Information technology (IT) plays a key role in processing of raw data into useful information. This is due to the fact that the amount of information that is needed for the smooth achievement of various objectives is continuously increasing. The world of modern business is increasingly complex, and everything that could be achieved using simple methods – has already been achieved. Therefore, it is necessary to formulate and solve increasingly complex tasks in order to achieve success in the competitive struggle. Meanwhile, as the situations that need to be addressed in the management process are becoming more complex – more and more information is needed to make the right decisions bringing beneficial outcomes. This increase in demand for information is very fast, sometimes even exponential. As the amount of information readily available grows much more slowly with the increase in the complexity of tasks to be solved, the effect of the *information gap*[2] increases, and it can be overcome *only by using computers* (Fig. 1.3).

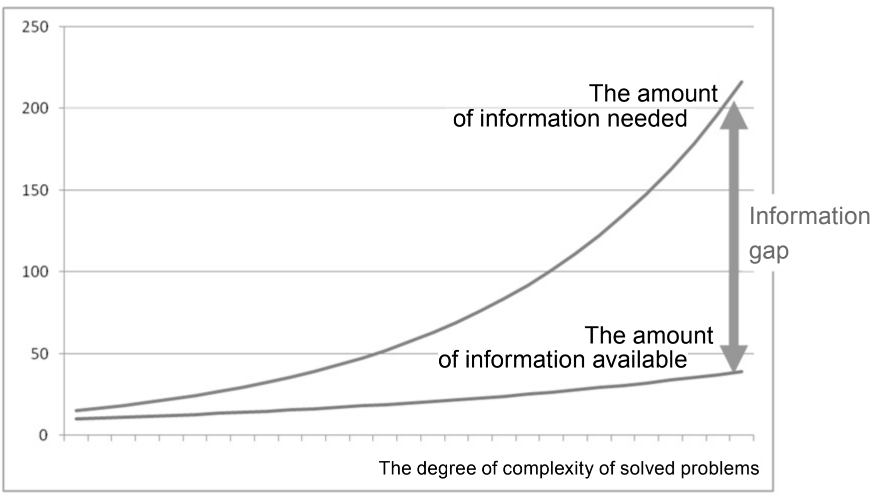

Fig. 1.3 The problem of an information gap
Source: Own work

Even the best information is not in itself a guarantee of effective action, because *knowledge* has to be the basis for making decisions. Knowledge is information selected, interpreted and used in such a way that together it forms a coherent image of reality, which is useful to understand what is currently

[2] The problem of an information gap is also referred to in Section 1.4.3.

taking place in this reality (for example, what is the state of the company and where is it heading, what are the signals coming from the market, what will be the effects of a given change in the law, what can be predicted from the political changes taking place, etc.)

These is a well-known saying, which summarises the relationship between information and knowledge very accurately:

> *Knowledge is made of information, like a house is made of bricks. However, not every pile of bricks is a house and not every aggregation of information creates knowledge.*

It is believed (and rightly so at the current stage of development of the computer science) that, as computers can (and should!) be used for converting data into information, the formation of the knowledge needed from information occurs mainly in a human mind (Fig. 1.4).

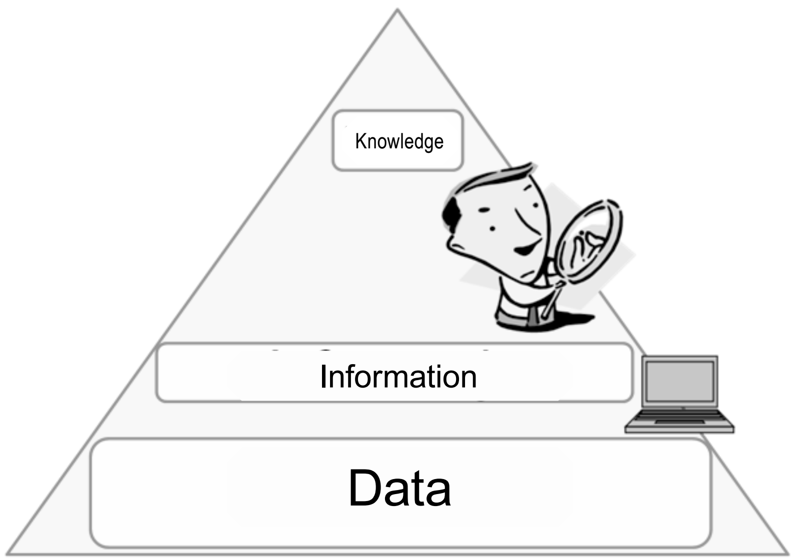

Fig. 1.4 The division of roles between computer technology and man in the processing of data into information and information into knowledge
Source: Own work

However, as the capabilities of information systems grow, particularly with the development of so-called artificial intelligence (Lula et al., 2007), the boundary between the range of activities available for computers and the area reserved for man's activities will shift. This will be more fully described in Section 2.5., but we can already say that new generations of computer systems may be capable of independent (automatic) generation of at least a semblance of knowledge. We should not, however, draw the conclusion from this that men will become useless, and machines will completely take over the process of managing companies. This will not happen, because, contrary to what Fig. 1.2 and Fig. 1.4 show, the process of wise use of data by no means stops at the stage of generating the necessary knowledge. One more step can be taken, which confirms that *wisdom is the real purpose of the whole process*. People need to make wise decisions, be guided by wisdom in setting long-term objectives and devising strategies for achieving these objectives. As we can see in the chart shown in Fig. 1.5 – wisdom arises if knowledge is complemented with the understanding of the principles which cause the phenomena and processes to take a particular course.

Analysing the construction of this chart, we can come up with useful conclusions. As you see, the first axis of the chart shows the intellectual depth of the considered objects (data, information, knowledge and wisdom), and the second one is the so-called contextual independence, which is the degree of generality (universal usefulness) of a given object.

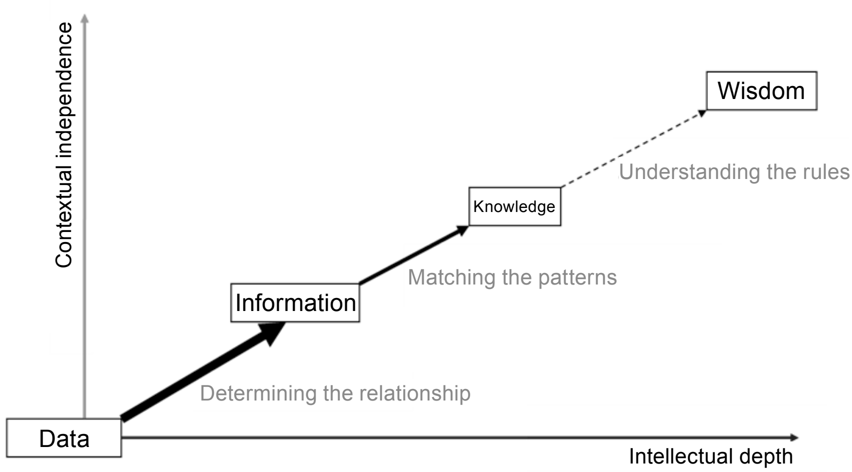

Fig. 1.5 Dependencies between data, information, knowledge and wisdom
Source: Own work

This system of coordinates starts with *data*. They have the intellectual depth of zero (a fact is noted, but it is not accompanied by any thought), and the contextual independence of zero, too (because data always relate to the specific fact and refer to it only).

By establishing relationships between data, through their processing and analysis, we can obtain information. Information has a greater degree of contextual independence (reflecting the general rules, not so tightly linked to a fact, as in the case of data), and it is a little deeper intellectually, because it is the result of data analysis – although this analysis might be quite shallow at this stage – for example, dividing the income according to the places where it was generated or analysing the tendency to variability over time (whether a given phenomenon increases or decreases). Activities of transforming data into information can be computerised, as illustrated by the thick arrow on the chart.

By searching for higher order correctness and regularity in information, and by matching certain schema, patterns, and models to the gathered information, it is transformed into *knowledge*. It already has a very high level of contextual independence (sometimes we even speak of a general knowledge – in contrast to detailed information and data strongly associated with facts), and is highly saturated with intellectual elements. As mentioned above, knowledge is mainly formed in human mind, but the new generation of computer systems (such as systems known as *data mining*) may facilitate and improve man's actions in this regard. This fact is marked in Fig. 1.5 with a thin arrow.

The next shift – from knowledge to wisdom – involves the *understanding of rules*. It is not enough to describe the world (which information and knowledge are used for). It has to be understood, because only then it can be effectively changed as needed. This is what true wisdom is, and currently it is the exclusive domain of man. However, it is difficult to exclude the possibility that new generations of computer systems, so-called cognitive systems (Ogiela and Tadeusiewicz, 2009) can also support human activities in this areas, which is indicated in Fig. 1.5 with a thin dashed line.

Another concept that is important for the understanding of topics discussed in this chapter is the concept of a *system* and a general systems theory (GST), which has a big impact on the MIS methodology. The authorship of GST is attributed to L. von Bertalanffy, and the research methodology which evolved from it is called systems methodology (Bertalanffy, 1968, pp. 32, 253, [after:] Bertalanffy, 1972, p. 411).

General systems theory is a logico- mathematical field, whose task is the formulation and derivation of those general principles that are applicable to "systems in general". In this way, the exact formulation of terms such as wholeness and sum, differentiation, *progressive mechanization, centralization, hierarchical order, finality and equifinality, etc., become possible, terms which occur in all sciences dealing with "systems" and imply their logical homology.*

General systems theory is an attempt at a formal recognition of the epistemological approach, ever-present in European philosophy. One of the first definition of a system had already been proposed by Aristotle. He argued, emphasising holistic and teleological aspects of a system, that (Bertalanffy 1972, p. 407): *The whole is greater than the sum of its parts.*

L. von Bertalanffy and other authors writing about the systems methodology, including Aristotle, belong to antireductionists, or holists (Flakiewicz 2002, p. 3). Holism in the sense of systems methodology is based on the assumption that the components of the system behave differently when they are isolated from each other and from the environment where they normally function, thus the system components should be always considered in its entirety, in their natural environment.

Although GST has been developing for almost 70 years, it has not yet work out a generally acceptable definition of a system, however, it is widely accepted that the main characteristics of a systems methodology are (Flakiewicz, 2002, p. 4):

- System is characterised by the purposefulness of action as defined by its creator
- The purpose of the system can be implemented in various ways
- No part of the system is isolated in relation to other parts of the system
- The system structure consists of parts with relations taking place between them

The first two statements emphasise the teleological dimension, the other two – the holistic dimension and this is why holism and teleology have to be assumed as the two most important attributes of systems.

Systems theory has become a methodological paradigm for many of the natural and exact sciences such as physics, chemistry and computer science. With time, it also began to penetrate into social sciences (sociology, psychology, political science) and management sciences. H. Simon was one of the strongest proponents of the use of general systems theory in management science. In his book entitled *Administrative Behavior* (Simon, 1997), first published in 1947, he defined an company as:

Adaptive system of physical, personal, and social components, which are held together by a network of intercommunications and by the willingness of its members to co-operate and to work towards common goals.

At the end of the discussion presented in this paragraph, it should be noted that the term 'system' is used very often in different contexts and it is intuitively understood by most people. Encyclopaedias and dictionaries indicate the etymology of the term 'system'. It (supposedly) originates from the Greek word σύστημα, which means composed of many components. This is in large part the essence of modern understanding of a system. It is a set of many elements connected to one another in such a way that, due to the synergy between these elements (their cooperation, mutual assistance and complementarity), the resulting whole is more valuable than indicated by the simple sum of the values (usefulness) of its components. Such an object is the main area of interest of information systems.

1.3. OVERVIEW OF THE DEFINITIONS OF INFORMATION

Information is one of the fundamental concepts – not just in computer science, but in the description of the world in general. Traditional science (physics, chemistry, astronomy, biology) described the world using two main components: *matter*, of which real world objects are made, and *energy*, which causes that these objects may be subject to many changes. However, the development of social sciences (including economics) and technology (especially automation, telecommunications and information technology) forced the description of the world to include one more equivalent element – *information*. In the previous section, we discussed the relationship between the four basic concepts that are used by man in the process of exploring the world. However, the concept of information has been given the most attention so far in the literature. It has even its own separate field of science – information science. Therefore, this section is devoted to the more formal description of the concept.

The term *information* is central to many areas of science. It also plays a key role in the ongoing socio-economic development. The electronic revolution, heralded by some visionaries, such as I. Baron and R. Curnow (1979) or A. Tofler (1986), that resulted in the creation of the information society became a reality (Barron and Curnow, 1979; Tofler, 1986, Castells, 2007), and information is now the most critical resource of modern businesses.

Despite its enormous importance, the concept of information does not have a single, universally accepted definition. There is an opinion that it is such a basic, general and ambiguous term, that any attempt at formal definition is not possible (Unold, 2004). Instead, it is possible to propose an operational or functional definition i.e., the one that puts the concept of information in the context of field-specific discussions – in the case of this textbook, this is the context of management information systems.

As mentioned above, information is a hardly definable term. The main reason for this is the interdisciplinary context of its use. In physics, the concept of information is used as a measure of organisation (entropy decrease). In telecommunications, information refers to the probability of a particular character being sent through a communication channel. In management science, we speak of managerial information and its importance in effective decision-making; in biology, this is genetic information; in linguistics, the importance of specific linguistic forms is considered. In library science, we speak of scientific information. In engineering sciences, information sent by a set of sensors is analysed, in general systems theory, a generalised category of a system processes generalised information.

Some authors disagree on the issue of a need for an accurate definition of the concept of information, while the other are calling for a precise definition. Others, due to the fuzzy nature of the concept, are quite satisfied with its intuitive understanding. The concept of information is understood intuitively by T.H. Davenport and M.L. Prusak (2000, p. 3), who make the distinction between data and information. These authors, claiming that: *information is data that makes a difference,* are followers of the concept of G. Bateson (1972), who claimed that information is: *a difference that makes a difference*.

It should be noted, however, that adopting intuitive definitions or, as R. Capurro and B. Hjørland (2003) called them – persuasive definitions[3] entails some risk. These authors report that, for example, acceptance of such a definition in the information science led to the creation of more than 700 definitions, which only caused a conceptual chaos. Quoting A.M. Schrader (1983, p. 99), the reasons for this include: uncritical citing of previous definitions, combining theory with practice, narrow perception of technology, incorrect analogies or formulation of recursive definitions. On the other hand, the notion of information, due to its original character, cannot be defined using semantic categories only. It is also necessary to include praxeological (pragmatic) aspects (Sundgren and Steneskog, 2003, p. 15).

The concept of information began to widely penetrate many disciplines starting in the late 40s of the last century. This is when the work by CE Shannon (1948) was published where a formal attempt to define this concept was made for the first time. The concept proposed by Shannon was so universal, that it quickly found use in the various scientific fields becoming the central part of a scientific discussion, which was surprising even for Shannon himself (Ritchie, 1986, [after:] Losee, 1997).

[3] Designed to impress the reader.

Information was the centre of research in cybernetics[4] (Ashby, 1957; Tadeusiewicz, 1994). N. Wiener is an author of a well known statement where he says that (Capurro and Hjørland, 2003):

information is information, not matter or energy,

thereby indicating that matter, energy and information are essential components of every system (Mynarski *et al.*, 1989, p. 146). One of the most frequently quoted definitions of information has been formulated by N. Wienerin cybernetics. According to it (Wiener, 1954, p. 16),

information is a name for the content of what is exchanged with the outer world as we adjust to it, and make our adjustment felt upon it.

One of the other field-independent proposals for the definition of information is the definition proposed in the work (Losee 1997, p. 258). R.M. Losee defines information in the context of general systems theory, and believes that:

information is produced by all processes and it is the values of characteristics in the processes' output that are information.

The definition proposed by Losee is similar to the definition of H. Greniewski, who defines information as Kempisty, 1973, p. 155):

distinguished states of the inputs and outputs of the system.

It seems that defining the concept of information in general terms is not justified. We then obtain a definition which is indeed true, but general enough that it is essentially devoid of any significant cognitive value. However, the precise analysis of the concepts in the context of the discussion is needed, resulting in the adoption of a field-specific definition, taking into account the subject of interest of particular fields of science, their methodology, language and conceptual specificity.

Analysing a great variety of definitions existing in literature, we can notice a certain division, significant from the perspective of our textbook. This division is in the form of a question: *Should we associate the concept*

[4] Cybernetics has a strong influence on the shape of modern science. This influence is manifested in the fact that, starting from the observation of living organisms, proposals for information processes occurring in them are formulated, to be then applied in different areas of creative human activity. The importance of cybernetics also consists of the fact that the themes developed in it are applied in all fields of science dealing with organised systems, such as machines (technology, particularly automation and electronics), organisms (physiology, neurology, biology, etc.) and society (sociology , economics, etc.) (Mazur, 1967).

of information with the necessity for the existence of man, or at least a certain system of interpretation, or, on the contrary, should we abstract from the consciousness of the recipient or the need for interpretation? (Capurro and Hjørland, 2003). One side shows definitions taking this necessity into account, the other shows definitions that reject it[5]. The list of basic attributes of the opposite views discussed and some authors presenting these approaches are presented in Table 1.1. Between these opposite views are concepts which attempt to reconcile these two extreme approaches, and the central position is occupied by a field-independent universal approach.

Table 1.1 Two basic approaches to the understanding of information

Approach	Qualitative	Quantitative
Main emphasis	Semantic and pragmatic aspects	The concept of measurement and order (decrease of entropy)
The need to consider the meaning and context	Yes	No
The presence of a man in the process of interpretation	Necessary	Unnecessary
Distinguishing between the concepts of data, information and related terms	Yes	No
Scientific fields and disciplines	Linguistics, psychology, sociology, economics, management science, information science, library science, information systems, (...)	Biology, genetics, chemistry, physics, telecommunications, tele-transmission, mathematics, computer science, (...)
Scholars	R. Ackoff, F. Machlup, T. H. Davenport, B. Langefors, B. Sundgren, P. Beynon-Davies	C.E. Shannon, W. Weaver, H. Nyquist, R.V.L. Hartley, J. von Neuman,

Source: Own work

The first of these opposite views is related to the perception of information typical to social sciences and humanities. Because of the need for an interpretation system, a human, it is referred to as an anthropological or subjective approach. One of the characteristic traits of this approach is the distinction between the concepts of data and information and related terms.

[5] Mazur (1970, p 18) notices yet a third group of publications i.e., those which, despite the fact that information is central to their deliberations, do not attempt to define this term at all.

The other view is appropriate to exact sciences, especially natural sciences and technology. In this approach, the concepts of data, information and many related concepts are not distinguished and are used interchangeably. The representatives of this movement, apart from the question of the meaning or context of information, do not deal with issues of interpretation. In this meaning, this trend is sometimes called an objective or naturalistic approach. In the deliberations of the authors presenting the objective approach, much attention is paid to measuring the amount of information. This is why it is often called a quantitative approach. To differentiate, the anthropological trend is often called a qualitative approach.

For obvious reasons, the topics related to the concept of information are frequently and widely raised in the literature in the field of management science, and information systems in particular. Unfortunately, there is no unanimity in the understanding and definition of this basic term (Checkland and Holwell, 2002). It should be noted, however, that because of the social nature of the above-mentioned disciplines, they are predominated by qualitative concepts. Some of them focus on semantic aspects, and some on pragmatic ones, and the second group is predominant.

In the following part, we will successively discuss the quantitative approach, with particular emphasis on the theory of C.E. Shannon, and the qualitative approach, focusing on the infological theory of B. Langefors and B. Sundgren. Subsequently, we will present the discussion on the role and importance of information in the process of managing an organisation.

1.3.1. Quantitative approach

The essence of the quantitative approach, as mentioned above, is to identify the concepts of *data* and *information*. It is assumed that data clearly define the content transferred, which is the basis for stating their objectivity – they cannot, in fact, be interpreted in many ways. For example, in mathematics, physics, or chemistry, the importance of basic terms, sums and values is indisputable. As stated by one of the most eminent authors presenting the quantitative approach, C.E. Shannon (1948, p. 379):

> *The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have meaning; that is, they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem.*

Literature on the subject, referring to the concept of *information theory*, repeatedly quotes the theory formulated by C.E. Shanonn (1948). This theory is the flagship perspective of the quantitative approach, so it will be discussed in detail below.

As previously mentioned, although there is a need for a definition of information, this concept is difficult to define. However, because a precise definition of information and its quantitative measure is very much needed nowadays, it does not mean that such definitions and such measurements are easily obtained. Hundreds of articles and books are available on the subject today, but one of the contributors to this textbook can recall what he himself wrote in a book published by Cracow University of Economics in 1974 (Kulik and Tadeusiewicz, 1974, pp. 289-290):

> *Difficulties in defining the concept of information result from the fact that information is multifaceted and it is difficult to decide which aspect should be the basis for the formulation of a definition. Information has a specific volume, expressed by the required number of alphabet symbols or signals in the technical equipment, necessary for its transmission. It has a specific meaning, usually expressed by a change of action of the recipient. It also has a certain value consisting in the fact that certain information is of great significance to the recipient or sender, and other is common and unimportant.*

Although these words were written 40 years ago, they have lost none of their topicality. Still there is no definition of information that would exhaust all its features and properties – and I think we will not live to see such a definition.

However, we have a good measure for the amount of information which, although formulated a few decades ago, is still valid and convenient. The author of this measure (and the related theory) – which proved to be one of the most important theories in history – was an American engineer named C.E. Shannon.

Shannon's brilliant idea was that instead of trying to define information directly (as previously was unsuccessfully attempted by other famous scholars, such as H. Nyquist and R. Hartley) – he decided to define the measure of a lack of information, and to express information as a factor that removes this lack of information. Seemingly this looks like trying to reach the right pocket with your left hand, but it turned out that this is an appropriate and effective method. In detail, Shannon's idea looks like this (Ogiela and Tadeusiewicz, 2009):

Imagine that you have received a message *M*. How much information does this message introduce? Looking at the message itself, we will not be able to answer this question, because the message itself can be very valuable to one recipient and worthless to another. Therefore, we must refer to what exactly the message concerns and what meaning it has for us. Imagine that we are interested in some event *E*, and that the message *M* is to inform us about this event. In this case, we need a mathematical formula describing the dependence *I(M/E)*, allowing us to estimate the amount of information *I* introduced by the message *M* about the event *E*.

Note that the formula *I(M/E)* has to make the amount of information dependent on both the message *M* and the event *E*. Dependence of information on the message *M* is obvious, but the fact that we focus on the dependence of the amount of information on the event *E*, which the message relates to, arises from the fact that the message *M* may contain additional information on completely different topics. For example, if we receive a message about the change in the exchange rate over the phone, then next to the event *E* (change of the exchange rate), the message *M* will contain additional information about the caller's sex, and if we know the caller's voice – also about their identity. However, we are only interested in information about the event *E* i.e., the change of the exchange rate, and any other information does not matter, so this is explicitly indicated in the formula sought.

The amount of information we can get about the event *E* depends on the degree of our *uncertainty* about this event. Imagine that this uncertainty can be estimated, and the function of uncertainty is denoted as *H(E)*. If the considered message M was able to completely eliminate this uncertainty, we feel that the amount of information we have received in the message *M* will be equal to the uncertainty removed:

$$I(M/E) = H(E) \tag{1}$$

Unfortunately, it does not always happen that after receiving the message *M* our uncertainty about the event *E* is reduced to zero. On the contrary, the rule is rather that we are still uncertain after receiving a message. Therefore, let us use the function *H(E/M) to* denote the uncertainty about the event *E* that remains after the receipt of the message *M*. It is therefore correct to rewrite the equation (1) in a more general form:

$$I(M/E) = H(E) - H(E/M) \tag{2}$$

Equation (2) shows that we choose to assume the degree of the *reduction* of our initial uncertainty, that occurred as a result of receiving the message *M, as a measure of the amount of information I(M/E).*

Formulas allowing for the determination of the uncertainty will be introduced a little further, but here we can "in advance" ensure that every measure of uncertainty is positive (or, in a special case, it can have the value zero), and it cannot be negative. This means that in particular

$$H(E/M) \geq 0 \tag{3}$$

which means that the case described by formula (1) sets out our *limit* (maximum) amount of information that we can obtain in the situation described here. This limit amount of information is determined by our initial uncertainty *H(E)*, and it is independent of the content of the message *M at this stage of the deliberations*.

The message *M*, however, is not without significance. According to formula (2), a poorly constructed message can make the acquisition of this limit amount of information impossible and we will have to settle for the amount of information which is

$$I(M/E) < H(E) \tag{4}$$

Note that a situation is not ruled out where the uncertainty after receiving the message *M* will not only not decrease, but the opposite – it will increase:

$$H(E/M) > H(E) \tag{5}$$

This situation occurs when the message *M* is actually a *disinformation*. In this case, the amount of information contributed by such a message is negative ($I(M/E)<0$) – and we have to deal with it.

The above discussion indicates that we will be able to measure (calculate) the amount of information when we find an effective method of calculating the measure of uncertainty *H(E)* and *H(E/M)*. Contrary to appearances, this is not at all difficult!

Following in the footsteps of Shannon's ideas, we link the measure of uncertainty regarding the event *H(E)* with the *probability* of this event *p(E)*. The relation between *H(E)* and *p(E)* is as follows:

The smaller the probability, the greater the uncertainty.

Referring to the example with exchange rates mentioned above, we see that the probability of a dramatic increase in the exchange rate (e.g. of euro) after a strong decrease today, yesterday and the day before is usually small. Thus, the uncertainty of whether we can speculate for an increase of the exchange rate is very large, and a message telling us that such a rise actually took place brings in a lot of valuable information. And *vice versa*: if for some reasons (for example, an annual financial report just published) it appears that the likelihood of tomorrow's rise of the exchange rate is very high, the uncertainty related to this is small, and the message that this expected rise took place carries little information.

Based on these considerations, we can formulate the first postulate for the measure of uncertainty sought – that it decreases as the probability increases. We can formally note this by introducing two events into the considerations: E_1 and E_2: one with a greater, and the other with a smaller probability:

$$p(E_1) > p(E_2) \tag{6}$$

In accordance with the formulated postulate, the inequality of the uncertainty will be in the opposite direction:

$$H(E_1) < H(E_2) \tag{7}$$

The discussed postulate can be complemented by moving to the limit value. The limit value to which the probability may increase is the value of 1, assigned to *certain* events. If so, then the *uncertainty* of an event with the probability of 1 should be zero – and this is another postulate, which can be applied to the definition of this concept.

$$p(E) = 1 \leftrightarrow H(E) = 0 \tag{8}$$

The third postulate is related to situations where we have to define the measure of uncertainty for a *complex event*. Suppose we are interested in the event E, consisting of the simultaneous occurrence of two *independent* events. For example, we expect a situation where the dollar exchange rate will fall dramatically and at the same time the shares of the PKO bank will rise. Shannon postulated that the uncertainty of component events simply be added in such a situation. This is the easiest and most convenient!

Let us see the result of this action. In the case of independent events, the probability of the occurrence of such a complex event E is (as we know) equal to the *product* of the probabilities of component events E_1 and E_2:

$$p(E) = p(E_1)\, p(E_2) \tag{9}$$

Another dependence results from the Shannon's postulate:

$$H(E) = H\,(p(E_1)\, p(E_2)\,) = H(E_1) + H(E_2) \tag{10}$$

There is only one function, which converts a product into a sum: *a logarithm*. Thus, the postulate (10) implies that the function defining the measure of uncertainty of a given event should be defined as the logarithm of probability. It is easy to see that with such a definition, the postulate expressed by the formula (8) is also true – we could say automatically – because the logarithm of one is zero.

A bit of trouble occurs in the context of a postulate expressed by formulas (6) and (7). The logarithm is an increasing function, so for larger values of the argument (probability), it will have higher values – and the opposite should be true. But there is a simple solution: we will use the *minus* sign before the logarithm, therefore, when the probability value will increase and this will be followed by the increase in the value of the logarithm – the value of uncertainty will be reduced. And that is the point!

So we have a definition for the measure of uncertainty in its basic form. Here it is:

$$H(E) = - \log_B p(E) \tag{11}$$

Still open is the matter of units, in which we will express the uncertainty. Note that if we select units to express uncertainty – exactly the same units will express the measure of the amount of information. This is related to the structure of formulas (1) and (2) from which this identity of units results.

Definition of a unit for the measure of uncertainty (and the measure of the amount of information) is closely linked with the symbol *B,* which appeared in the formula (11) as the basis of the logarithm, which we want to use. When Shannon created the foundations of his theory, decimal logarithms, precursors of modern calculators, were in common use. Therefore, in the first works, $B = 10$ was assumed, and a unit for the measure of the amount of information was created on this basis. This was called the *decimal information unit – dit in short*. Later, mathematicians who appreciate the advantages of natural logarithms i.e., those where $B = e = 2.7182818...$ took up the study of the basics of information theory. After introducing this basis of logarithms, another unit appeared, defined as the *natural information unit – nit in short*.

However, the development of computer technology and the related development, the "career" of the binary system meant that $B = 2$ was increasingly used, resulting in that the unit known as *binary information unit – bit* in short, became popular.

Perhaps some readers will be alarmed at this point – after all, a bit in computing means something completely different!

This is a completely unfounded concern, as everything is correct here. The name bit in computer technology is referred to that part of a microprocessor, register, or digital memory, which can store either 0 or 1. Also, when sending information, the concept of a bit is used to determine a portion of the signal which may be a zero or a one.

Let us consider how much information (in accordance with the above definition) can be stored or moved by such a system element where two states are possible. The initial probability that this element will be zero (or one) is exactly 0.5. Thus, the uncertainty of the event, consisting of entering this or another digit to this element is:

$$H = - \log_2 0.5 = 1$$

So, the amount of information, which is located in one bit is exactly 1 bit. Therefore, these traditional meanings of the word bit are not in conflict with the above-defined unit of the measure of the degree of uncertainty and the measure of the amount of information and everything is in concert!

We already know how to calculate *H(E)*. And what about *H(E/M)*?

Exactly the same way, but different probabilities are taken into account – instead of the a priori probability *p(E)*, we have to insert the conditional probability *p(E/M)* into the appropriate formula – i.e., the probability of an event *M* after the receipt of the message *E*.

It is time to introduce an important term into our deliberations.

The measure of uncertainty *H(E)* and all measures related to it are called *entropy.*

This concept was introduced by physicists to explain the irreversible nature of some of thermodynamic processes. Physicists have found that all processes in nature proceed in such a way that the entropy is constantly increasing. Discussions on this subject in physics led to very far-reaching conclusions, for example, they have predicted that the Universe where we now live is doomed, because the life-giving light of the Sun (and other stars) will expire in the future due to depletion of stocks of nuclear fuel inside them. Since the process of energy squandering, which is the norm among stars and galaxies, is irreversible (because it is characterised by the increase of entropy), the "thermal death" of the entire cosmos is inevitable. Entropy also allows to explain many phenomena in biology, including basic questions about the nature of life and death. So it was a great scientific stir, when the research related to the measure of the degree of uncertainty in Shannon's information theory showed that the measure he introduced is essentially identical (with an accuracy of the scaling factor) with entropy. There is no room here for a lengthy discussion on this topic, but it is worth mentioning that for example, the process of spontaneous destruction and disintegration of old buildings is unavoidable, because it is associated with an increase of entropy, which is the opposite of information (in this case, information contained in the structure of buildings). We do not observe bricks laying themselves and forming a new house – because then entropy would decrease, and this is impossible. Similarly, a message passed from mouth to mouth is distorted, but it never happens that when a message is transmitted and repeated, there is more and more sense and real information in it – because the law of the increase of entropy applies here too.

But let us return to the theory of information and analyse the following problem, of importance to practical computer science:

In the considerations above, we have obtained estimates of the maximum amount of information we can expect in the message M about the event *E*. However, this estimate, expressed in the formula (1), referred to the entropy of the event *E only*, not linking this maximum amount of information to the characteristics of the message *M*. This is disadvantageous because we have no influence on the events the messages relate to, but we do have control of the way of formulating the messages. We would prefer to know what to do with a message so that it will introduce the greatest amount of useful information. In order to achieve this, we must use a certain simple identity known from the probability theory and apply it to entropy.

This identity shows that the cumulative probability of the simultaneous appearance of a specific message *M* and event *E* can be expressed in two ways:

$$p(ME) = p(E)\, p(M/E) = p(M)\, p(E/M) \tag{12}$$

Note that we had to use conditional probabilities in this formula, because in contrast to the situation described by the formula (9) – we cannot use the model which applies to independent events. The event *E* and the message *M* regarding this event are objects which are dependent very strongly on each other, because this is the essence of the message.

Using the analogy with formula (10), we can rewrite the identity (12) in the form:

$$H(E) + H(M/E) = H(M) + H(E/M) \tag{13}$$

By moving both conditional entropies to the other side, we get another identity:

$$H(E) - H(E/M) = H(M) - H(M/E) \tag{14}$$

Note that the identity on the left (13) is identical to the expression appearing in formula (2), which therefore can be rewritten as:

$$I(M/E) = H(M) - H(M/E) \tag{15}$$

This formula says something important and useful in practice: the largest amount of information that can be included in a message is limited by the a priori entropy of the message *H(M)*. The *H(M/E)* component may reduce this maximum (theoretical) efficiency of information, but it cannot increase it. Incidentally, let us see what the *H(M/E)* component actually means. It is a measure of the degree of uncertainty as to the message M when the event E, which this message is to inform about, will be completely known. A non-zero value of *H(M/E)* means that the system generating the message is defective: it is unclear what the journalist will write after we know exactly what really happened; it is unknown what signal the sensor will send, although we know the value of the measured parameter. Therefore, the non-zero value of *H(M/E)* is an embarrassment to the information system. However, since a non-zero value of *H(M/E)* can occur – we should write the dependence:

$$I(M/E) < H(M) \tag{16}$$

This dependence results in the following recommendation: To transfer as much information as possible, the greatest possible initial entropy of the message must be ensured.

How, in practice, can we fulfil this recommendation?

To answer this question, let's examine the structure of a message.

There are very many different forms of representation of a message (for example, a text, a table with numbers, a map, an image, sound, video, etc.), but each of these forms can be reduced to a single model: a message is a sequence of some *symbols*. Still, we will be considering the model of a text message, because this is the one you are looking at now and it will be the easiest to imagine the matters discussed here, but the conclusions we will formulate will apply to all forms of messages. Anyway, in the era of computers and the Internet, all messages eventually take the form of a series of ones and zeros...

Let us consider the symbols making up the text – that is, letters. The process of reading a text consists of a sequence of events and each event consists of reading one (successive) letter. What is the uncertainty as to what is the next letter which we will read and what is therefore its entropy?

Different letters have different probabilities, so we have to consider several possibilities. We will number the letters of the alphabet consecutively from 1 to N, and assign the appropriate probability to them:

$$p_1, p_2, \dots , p_N \tag{17}$$

The sum of these probabilities must be 1, since it is certain that some letter will be read:

$$\sum_{i=1}^{N} p_i \tag{18}$$

Uncertainty resulting from the formula (11) is related to each of these letters, so we have a set of values of entropy in the form of the sequence:

$$-\log_2 p_1, -\log_2 p_2, \ldots, -\log_2 p_N \tag{19}$$

The entropy of each successive letter of the text is therefore a *random variable* with values given by the formula (19), and the precise values of this random variable are accepted with probabilities given by the formula (17). Since the entropy of a single character in the text is a random variable, then its proper assessment requires the use of a formula for the *expected value*. In this formula, different values that entropy assumes (19) should be multiplied by appropriate probabilities these values are adopted with (17) and the results summed up in the range used in the formula (18). As a result, we will get the expected value of entropy per one symbol in the text $H(s)$:

$$H(s) = -\sum p_i \log_2 p_i \tag{20}$$

With the estimated entropy per one symbol in the text and knowing the number of characters C appearing in the analysed message, we can estimate its entropy

$$H(M) = C\,H(s), \tag{21}$$

and this will allow us to estimate (in accordance with formula (16)) whether this message will be able to transfer the required amount of information.

The key issue appears at this point:

What properties should a set of characters used to create message M have in order to enable it to carry the required amount of information using as small number of characters as possible? Let us note that each character has a cost – when writing on paper, it is the cost of the piece of the page occupied by this character. When stored in the computer, each character occupies a part of memory. When sending over a network, each character extends the transmission time. Therefore, our objective at this point is subject to the principle:

The more information one character can convey the better.

When will one character transfer more information?

When its entropy given by formula (20) will be as large as possible.

How can we increase the entropy described by formula (20)? It could be proved mathematically, but the proof is long and difficult. But we can also use common sense, by considering when our uncertainty about what will be the next character in the text is the largest? Of course, it is when each character can appear with the same probability! Any preference reduces uncertainty, because when some characters are more likely to appear than others – this facilitates the task for someone who wants to guess these characters. Such unequal probabilities are "hunted" by spies trying to crack a cipher that protects classified information.

Exactly the same conclusion comes from the mathematical analysis of formula (20). It will assume the maximum value when

$$p_1 = p_2 = \ldots = p_N \tag{22}$$

When we also consider the dependence (18), it becomes obvious that for each character number, there must be a dependence

$$p_i = \frac{1}{N} \tag{23}$$

By substituting the dependence (23) into equation (20), we receive an estimate showing what is the maximum entropy per one symbol when they are all equally probable:

$$H_{\max}(s) = -\sum p_i \log_2 p_i = -\sum \frac{1}{N} \log_2 \frac{1}{N} = \log_2 N \tag{24}$$

Actual information systems rarely use such a way to record information that guarantees the highest entropy for each character used. For example, in Polish, where N is 32, it can be calculated from the formula (24) that

$$Hmax(s) = 5\ bit/symbol \tag{25}$$

However, the letters in Polish have different probabilities, as illustrated in Table 1.2. This table does not include Polish diacritics i.e., *ą* is equated with *a*, *c* with *ć*, *ę* with *e* etc. Also, the characters *z*, *ź* and *ż* are all thrown in to one "bag". This made it possible to show the *unevenness* of the distribution of probability for various characters without creating an excessively large table.[6]

[6] The full table of probabilities of all letters occurring in texts written in Polish (including letters, which formally do not belong to the Polish alphabet, but sometimes occur in Polish texts, such as *q*, *v*, *x*) can be found, for example, at http://pl .wikipedia.org / wiki / Alfabet_polski.

If we insert these probabilities into formula (20), it turns out that the entropy of a single letter in the Polish text is below 2 bits per symbol.

Table 1.2 Simplified set of probabilities of Polish letters

s_i	p_i	s_i	p_i	s_i	p_i	s_i	p_i
a	0.080	g	0.010	m	0.024	t	0.024
b	0.013	h	0.010	n	0.047	u	0.018
c	0.038	i	0.070	o	0.071	w	0.036
d	0.030	j	0.019	p	0.024	y	0.032
e	0.069	k	0.027	r	0.035	z	0.058
f	0.001	l	0.031	s	0.038	space	0.172

Source: Own work

However, this is not the end. An exact consideration of the size of uncertainty of a given letter in a piece of text necessitates taking the context into account. Knowing the previous letters, it is easier to guess the next one. For example, seeing the inscription *KRAKÓ** we can easily guess that the final star replaces the letter *W*. We are sure of this, so the probability of the letter *W* in this place and in this context is 1, while the probability of the same letter without context is only 0.036. As you can see, when determining the actual entropy of a single letter, the *conditional probabilities* should be taken into account, and an appropriately developed formula taking into account such conditional probabilities should be used instead of formula (20). Let us skip the details and look at the final result right away. The actual entropy of a single letter in a text written in Polish is:

$$H(s) = 1 \text{ bit/symbol} \tag{26}$$

Let's compare the values given in formulas (25) and (26) and consider what they really mean.

It turns out that an important feature of an information system was found here, the so-called *redundancy*, calculated from the formula:

$$R = \frac{H_{\max}(s) - H(s)}{H_{\max}(s)} \tag{27}$$

It is the measure of redundancy (in our case – of the Polish language), and it determines how much of the information, which could be carried by a single symbol of the language (25), is in fact not used due to the fact that the actual amounts of information transmitted by one symbol (26) are smaller. At first, substituting the data from formulas (25) and (26) into formula (27) gives a saddening result: 80% of the capability of conveying information is wasted in Polish, and the words, sentences and whole books provide information on the level 20% of what that they could as a result of the properties of our language. For example, this book could have five times fewer pages (and it could cost five times less!) if the full potential entropy of the Polish alphabet was used. The same applies to the storage of text on a computer disk or sending it over the network. Everywhere, the redundancy of Polish causes that we pay five times more!

The first question that comes to mind after reading this is: Does this property apply to our language only?

Of course, no, and a similar level of redundancy was found for *all* languages, both living and dead (Latin). Interestingly, a similar about-80% redundancy was found by analysing the quasi-linguistic communication systems, such as for example the African tom-toms. It turned out their beat carrying messages through the impenetrable jungle also contains about 80% of redundancy!

Of course there are slight differences between languages, e.g. English has less redundancy than Polish, which means that the Polish text, translated into English usually takes up less space. But Polish is not "the worst" when it comes to the amount of redundancy, because even greater redundancy is characteristic of most Romance languages, especially Portuguese.

Since all languages have so much redundancy, maybe it is needed for something?

Indeed – it is. But for what?

To answer this question, imagine that you sent a text (SMS) message and make a typo, writing PYRT instead of PART. It is highly likely that the recipient of the text will not even notice the mistake, and if they do, they will easily guess what the error was and read the word correctly. If, however, you enter a number – for example, PIN – 7278 (note: the same keys are used as in the word PART!) on the same keyboard and make the same mistake as above, sending in consequence the number 7978 – the recipient will have no chance to detect that there would be an error in the message sent, and certainly they will not be able to guess in any way what is the correct number.

What is the difference between the transmission of words and the transmission of numbers? When sending words, we use the fact that not all combinations of letters are valid words. This is so due to redundancy. Some combinations of letters do not fit into any existing word, so when they occur – this clearly indicates the presence of an error. And when we find that there is an error – we can take advantage of unequal probabilities of different letters in different contexts, which in turn allows us to determine what word should have been written instead of the one that appears. This mechanism works reliably in all languages, so we can properly read partially blurred or smudged words, we can correctly understand the word spoken imprecisely, an oral message can be heard even in the presence of strong noise. Redundancy ensures the *reliability* of language communication in a variety of linguistic distortions and interferences we deal with every day.

As opposed to words written in any language, the representation of numbers does not provide redundancy. Each digit is equally probable, so if there is a distortion of the message – nothing can warn us that the message has been distorted, and what's more, nothing can help us figure out what correct digit should appear instead of the incorrectly written one.

Two conclusions can be drawn from this discussion.

First, in computer systems, where any loss or mispresentation of any character is fully excluded, redundancy is unnecessary. If a natural way of expressing specific information provided such redundancy (for example, the message considered is a text written in Polish and has 80% of redundant information) – this redundancy should be removed before collecting information on a computer disk or before it is sent via a telecommunications link. And this is done using programs performing the so-called file compression. These popular programs, such as PKZIP or RAR, replace the original information which is characterised by high redundancy with its encoded form, where the probabilities of all bytes are practically identical, causing that less of these bytes is needed for saving the same information (text, image, sound, etc.).

But we can also imagine the opposite situation, when for some reason we expect that the information which is of interests to us will be disrupted and distorted. Then, to save or transfer such information which is especially vulnerable, special codes are used, which may introduce an arbitrarily large redundancy into the message. We can then detect and correct multiple errors, or properly play even very distorted images, so that communication can be effective and safe even at very high levels of noise and other distortions.

Often, especially in studies of an encyclopaedic nature, Shannon's theory is considered to be the (only) theory of information. However, note that Shannon, when defining the *amount of information* in his work, has not defined the concept of *information*. Besides, he emphasised only one of the attributes of information, namely its *novelty.* Paradoxically, a portion of information consisting of a sequence of random, incomprehensible characters has more informational value in the light of Shannon's theory than any well-formed sentence in English of the same length. Despite the fact that the main achievement of Shannon's theory is to enable the measurement of information, it is not possible in every case. It is necessary to define a finite set of events and the probability of occurrence of each of them. If it is impossible to determine these conditions, the measurement of information is not possible at all (Mazur, 1970).

While in no way diminishing the importance of Shannon's theory, which is without a doubt the most advanced theoretically and formally, generally accepted theoretical model of the notion of information, it should be noted that reserving the term *theory of information* exclusively for Shannon's concept is a misuse, especially that Shannon himself talks about the theory of communication in the title of his work. A more proper term would be a *theory of communication* or a *quantitative theory of information* (Mazur, 1970). Also, this position is taken by R. Capurro and B. Hjørland, who support the interdisciplinarity of the term, as seen from the point of view of information science, in one of the most comprehensive studies on the concept of information (Capurro and Hjørland, 2003).

1.3.2. Qualitative approach

Qualitative approach as opposed to quantitative approach distinguishes between the concepts of *data* and *information*. According to some sciences that perceive information from the qualitative perspective, particularly in information science and knowledge management, two other concepts are additionally distinguished: *knowledge* and *wisdom*. As already discussed in Section 1.2 (Fig. 1.2), there is a concept in the literature on the subject that makes it possible to illustrate the interaction taking place between these notions. This is called a pyramid or hierarchy of information. This structure is also known by the acronym *DIKW*[7], derived from the words *data*, *information*, *knowledge* and *wisdom*.

Data layer is the most elementary layer in the hierarchy and wisdom is the most general one, which is consistent with the universal and intuitive understanding of these concepts. Lower layer is the reference point for the definition of the upper layer. Thus, the basis for the definition of information is the concept of data, the concept of knowledge is defined on the basis of the concept of *information*, and *wisdom* on the basis of *knowledge*. Although no one is likely to question the general concept of the hierarchy itself and the direction of making generalisations, the DIKW hierarchy has many critics. The problem areas include its little practical significance and the lack of clear, satisfactory, and universally acceptable definition of the concepts themselves and of the processes that transform one into another (Rowley, 2007; Frick, 2009). Since the definition of the data layer is the foundation of all definitions,

[7] The authorship of the DIKW hierarchy is attributed to M. Zeleny (1987) and R.L. Ackoff (1989), who proposed its extended version (with the layer of *understanding*, located between knowledge and wisdom). This structure, in its extended form, has not found universal acceptance and the hierarchy of cognitive concepts still applies to four levels.

its erroneous determination causes that the error contained there is automatically replicated, and in many cases amplified in subsequent layers.

> *Very often data are defined as facts – for example, Davenport and Prusak (2000, p. 2) define this term as a collection of individual objective facts.*

Adoption of the above definition causes that all data are treated as true. But this is not consistent with reality, because the data often are burdened with more or less significant errors. These errors result from the imperfections of recording tools, or simply mistakes in their input. Adoption of such a definition directly affects the definition of the derivative concepts. If all data are objectively true, then the data interpretation process does not require their verification, which undoubtedly is also not consistent with practice. As a result of adopting the above recursive definitional scheme, the definition of a certain concept is even more ambiguous and susceptible to errors, the higher it is in the hierarchy. A more detailed discussion of the different components of the DIKW pyramid can be found in (Grabowski and Zając, 2009).

The main distinguishing feature of a qualitative approach is the assumption that the system exists where information appears at the output as a result of the interpretation of data. The interpretation of data, closely related to the concept of *meaning* of data is such a complex issue that the authors of the quantitative approach have ruled it out from the scope of their deliberations. The proponents of qualitative approach generally believe that man is the only system that can actually effectively implement the process of data interpretation. Therefore, the process of informing in the qualitative approach has an anthropological dimension (applies to a process that takes place in a human mind). Semantic analysis of the concept of *information can shed s*ome light on the understanding the essence of the concept of *interpretation*.

The etymology of the word *information* points to the Latin term: *informatio*. According to Capurro and Hjørland (2003), this term has two basic meanings: (1) molding the mind, and (2) communicating knowledge. The PWN Encyclopaedia, indicating the Latin term *informatio* as Capuro and Hjørland (2003), notes that information is an *idea*, a *notice* or an *explanation.* To understand the conveyed information, it is therefore necessary for the recipient to have the ability to interpret, including the ability to use a particular language, experience, acquired knowledge, education and cultural context – because only then they will be able to have a (correct) *idea* of the message. Please bear in mind the fact that *idea* is an imprecise term, for there is no guarantee that the recipient understands the information in the way desired by the sender. Lack of determinism in the communication process is further complicated by the fact that the process of interpretation is determined by many other factors that were not previously mentioned, such as the psychophysical condition of the recipient. This does not mean that communication between people is indeterminate or random, because we know from experience that, in general, we understand simple messages unambiguously. However, communicating complex content can and often does encounter various interpretation barriers. For the above reasons, information is subjective by nature according to the qualitative approach.

The most advanced and structured concepts of the qualitative approach, particularly useful in the field of information systems, include the infological theory developed by B. Langefors (1973) and B. Sundgren (1973), represented in Poland by, among others, M. Kuraś (1987), B. Stefanowicz (1987), W. Flakiewicz (2002) and J. Unold (2004).

As indicated above, the concepts of data and information are distinguished in the infological perspective, which is an example of a qualitative approach. In the case of the concept of data, two more subcategories are distinguished: *direct data* and *indirect data* (Sundgren and Steneskog, 2003). Direct data are a result of the interpretation of signals from the environment taken in through the senses by the human mind. Based on these data, the human mind, in an iterative process of continuous learning, creates models of real-world concepts for its use. The human mind can then try to express some of these concepts by means of symbolic data, being the indirect data. Some of these data are easy and relatively unambiguous in interpretation and some are much more difficult. The first group includes gestures, images or sounds of onomatopoetic nature. Others, such as words spoken in a specific language requiring the recipient's knowledge of the principles of interpretation, belong to the second group. Indirect data can then be interpreted by the same person (read from notes), or communicated in some form (e.g. a letter) to another person. Of course there is no guarantee that the data in the mind of the other person will create the same concept as described by the author using these data. Moreover, even the author of indirect data, reading them after a while might not read them the same way. Both direct and indirect data are interpreted in the human mind through the conceptual models generated in the learning process, which forms a kind of a frame of reference. The social environment where the person lives affects the final shape of this system.

The process by which a person interprets the data, hereinafter referred to as the information process, has been described by B. Langefors (1973) in the so-called infological equation. It is expressed by the formula:

$$I=i(D,S,t) \tag{28}$$

where I means the content of the information acquired by man, i – the process of interpreting and giving meaning, D – the data received, S – the frame of reference or accumulated (prior) knowledge used by a person who interprets, and, t – time when the process of interpretation takes place.

One of the fundamental postulates of the infological approach is the assumption that information is stored exclusively in human minds. It is generally distorted in the process of communication. As indicated by Sundgern and Steneskog (2003, p. 14), man has always tried to extend and strengthen the capacity of storage of information outside of its own memory during the development of the human civilisation. He used symbolic (indirect) data for this purpose, and a variety of ways to record them. Also, the processes of sharing and communicating information take place through the use of data and methods. Information and communications technology methods and means play an essential role in the process of recording and transferring data at the present stage of civilisation development.

In order for the transmitted information to be subject to the smallest possible distortion in the communication process, it is necessary to ensure that the frames of reference are similar in the minds of the sender and recipient. This will enable the recipient to accurately, or relatively accurately interpret data. This role is assumed by metadata, or data about data. They describe the rules for data interpretation. Metadata define the meaning of data, determine their accuracy, origin, format, and describe the processes performed on them (Sundgren and Steneskog, 2003, p. 28).

In the infological approach, as already mentioned, information can be processed exclusively in the minds of people. This means that we cannot consider the information processing in relation to computer methods and tools. Computers can at most process data. Therefore, authors presenting the infological approach distinguish between two important perspectives of a computerised information system: *infological* and *datalogical*. The infological perspective has already been discussed above. The datalogical perspective, subordinate to the infological perspective, applies to those elements of the information processing which can be performed technologically – so we can say that it is instrumental for the infological perspective. The datalogical perspective includes data, metadata, data processing rules described in procedures and algorithms, and the technical tools implementing them. The purpose of a well-designed and implemented datalogical layer of an information system is to provide data with sufficient quality and metadata, so that the information provided or stored by the user of the system could be reproduced as precisely as possible in the mind of another user, or to allow such an interpretation of the data stored in the system, which will create new, never previously articulated information in the mind of the recipient.

A diagram of the information process from the infolocial perspective is shown in Fig. 1.6.

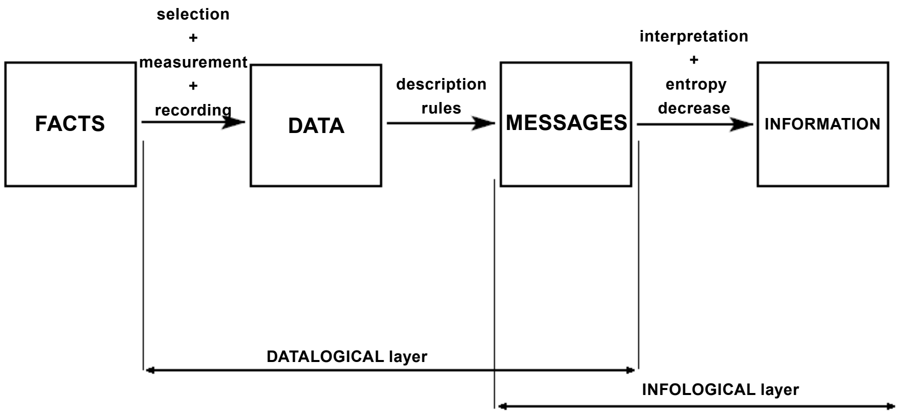

Fig. 1.6 The information process from the infological perspective
Source: Adapted from (Kuraś, 1987)

Facts are the fundamental element of reality in the infological perspective. Of course, only the ones that are initially relevant to the information process are recorded. They are then measured and coded using a variety of scales: nominal, ordinal, interval, or continuous or expressed as text descriptions, drawings, or other formats, such as multimedia content. The result is individual data, which do not include any content by themselves until they have been linked together using certain description rules. These rules are metadata. Metadata convert data into messages. Man, with the ability to interpret by the creative compilation of data, reads their meaning, which has a higher level of organisation than the content of individual messages. If a portion of a message with a specific, man-understandable meaning, introduces an element of novelty to his or her consciousness, that is, reduces his or her ignorance, then it becomes information. The whole process may be and usually is subject to some distortion associated with the improper selection of facts, which are to be recorded, improper measuring instruments and improper coding of data or with mistakes in recording and coding of data, incorrect determination of metadata, incorrectly operating algorithms correlating data and psychophysical predispositions of man. One can also relate these distortions to the existence of information filters: technical, semantic and pragmatic (Weaver, 1949[8], pp. 1-3; Kisielnicki and Sroka, 2005, pp. 22-23).

[8] Weaver (1949) was first to draw attention to three levels of communication: technical, semantic and pragmatic. They are related to, respectively, the accuracy, meaning and effec-

At the end of the discussion included in this section, it is worth noting that, although often portrayed as a competitive concept, the infological theory formulated by Langefors (1973) and Sundgren (1973) is not contrary to the quantitative theory of information. It may be presumed that, in a sense, it is the development of this theory. The concepts and achievements of the quantitative theory, such as the theory of codes, methods of error correction or data compression algorithms may be and are successfully used in the datalogical layer, and thus C.E. Shannon's quantitative information theory provides an excellent theoretical model for the construction of the datalogical layer in the infological model of the information process by B. Lagefors and B. Sundgren.

1.4. THE INFORMATION SYSTEM AS PART OF AN ORGANISATIONAL SYSTEM

1.4.1. Information and its importance in the organisation

The context of the field of information systems is an organisation, usually a business, focused on profits. Therefore, the main area of interest of MIS is the role that information plays in the organisation[9]. Information is one of the most important factors that condition the implementation of activities, which the theory calls the management functions. These include: planning, organising, coordinating, motivating and controlling. The essence of carrying out each of these functions is to make decisions. Therefore, the management process is often identified with the process of decision-making (Koźmiński and Piotrowski, 1997; Unold, 2004). The importance of the effect of information on decision-making is also indicated by J. Kisielnicki and H. Sroka (2005, p. 14). These authors, quoting J. Stoner, and Ch. Wankel (1992) argue that (1) data (raw facts) are at the input of the decision-making process, and

tiveness of information conveyance. As a promoter of the C.E. Shannon's theory, Weaver also pointed out that this theory applies to the first level only, but he believed it also has far-reaching implications for other levels, which are more important, slightly overlapping each other, and much harder to define.

9 Organisation and management topics are widely discussed in scientific literature. Here, we will discuss only those issues that directly concern the field of information systems (as it was defined in the introduction to this chapter) and at the same time are necessary, in the opinion of the author, to support the discussion. Extensive discussion on organisation and management can be found, among others, in Polish monographs in this field: Zieleniewski (1981), Stabryła (1995), Koźmiński and Piotrowski (1997), Obłój (2007).

after examining them, they can become (2) information, which in turn can become (3) management information, which actually are proposals for possible actions. Proposals for actions may lead to (4) decisions and, consequently, (5) implementation of actions.

A subprocess can be distinguished in the decision-making process described above, where (2) information is converted into (3) the management information, meaning a such one which provides conclusions regarding possible actions. (Management) information systems limit the concept of information to this kind of information only. When describing the attributes of management information, J. Unold (2003, p. 164) states that: management information is information that (1) is used to implement certain management functions, and (2) applies to all levels of management.

The importance of desirable features that determine the quality of management information is emphasised in the vast majority of discussions related to this topic. Although it is difficult to unequivocally define the concept of quality of information, it is often determined by the presence of a certain set of its desirable characteristics (Niedźwiedziński, 1987; Kisielnicki, 1987; Stefanowicz, 2007) or a proposition of a formal or quantitative assessment of the quality of information based on its characteristics (Stefanowicz, 1987).

M. Niedźwiedziński (1987) systematised a number[10] of more or less precisely defined features of information found in literature, and proposed a set of 35 characteristics: aggregation, timeliness, appropriateness, worthiness, decisiveness, accuracy, availability, efficiency, flexibility, quantity, unambiguousness, completeness, communicativeness, costliness, volume, cost-effectiveness, fullness, comparability, labour-intensiveness, authenticity, preciseness, prospectivity, suitability, assimilability, redundancy, retrospectiveness, prolixity, reliability, rationality, consistency, speed, value, credibility, faithfulness, originality. Due to lack of space, the definitions of these features will not be included. As mentioned above, as well as in the literature, they are not unambiguous, and their intuitive understanding does not differ substantially from the definitions proposed by M. Niedźwiedziński. At the level of generality of the discussion contained in this section, it seems that the reasonable thing is to present a smaller number of criteria, especially since some of them seem to have similar meaning. At a higher level of generality, sub-criteria can be grouped into more general categories. For example, worthiness and profitability in fact determine the value, and faithfulness and credibility relate to reliability. Research done by psychologists suggests that the limit of elements,

[10] When defining the system of quality assessment of information systems, J. Kisielnicki (1987) lists 19 sub-criteria, while 5 of them are independent of the type of a particular system and the remaining 14 depend on it.

which does not interfere with man's successful mastery of complexity is 7±2 (Miller, 1956), so it seems a good idea to limit the number of considered attributes of information to this particular number.[11]

B. Stefanowicz (2007) limits the number of information criteria to ten, drawing attention to the low precision and lack of accuracy in previous studies of this subject matter. The reason for this should be sought in definitional problems. Some of the features mentioned concerns the information itself and some concern the methods of its collection and interpretation. Stefanowicz believes that many problems can be eliminated using the infological perspective, and analyses seven selected qualitative features of information in this respect: timeliness, reliability, accuracy, completeness, unambiguousness, flexibility and relevance. At the same time, he emphasises (p. 73), that the list of quality features of information can be freely extended, and each of the features of information should be considered from the point of view of the criterion of usefulness. Each characteristic is subjective by nature, and its weight will vary in each specific case.

When speaking about the attributes of information, W. Abramowicz (2008) also believes that they are subjective. He divides the selected list of attributes, essential to the discussion, into two groups: informative and technical. The first group includes accuracy, timeliness, retrospectiveness, predictability, reliability, usability, completeness and assimilability, the second one includes volume and form related to the medium in which it is stored. In his deliberations, he also brings attention to relevancy. This attribute is defined as validity and appropriateness of the information assigned by the user. This characteristic is particularly important in the present age, when information is obtained by searching large data resources.

This textbook also assumes that the information criteria are subjective in nature. The question arises whether after narrowing the context of considerations, e.g. to decision support issues, we can try to specify a certain set of them, that will enable us to refer to the concept of quality of information in a sufficiently unambiguous manner. Such a useful set of information criteria, arbitrary in a sense, but satisfying the above requirements (i.e., widely

[11] Besides, the author (Niedźwiedziński, 1987), when determining the relationship between the proposed 35 characteristics, aggregated them, which allowed for selecting two groups of features: formal and substantive. In addition, he created new superior features: economic efficiency, ergonomic effectiveness and usability. The group of formal features does not contain any sub-groups, while the group of substantive ones includes four sub-groups: related to the time, meaning, scope and objectiveness. This gives a total of five groups at the lowest level of the hierarchy and is consistent with the number proposed by Miller.

accepted, with a limited number of criteria), has been proposed in the COBIT standard (ITGI, 2007)[12].

COBIT defines, inter alia: 7 criteria, which specify the requirements put on information from the perspective of organisational processes. They include:

1. Effectiveness, which relates to satisfying all requirements of the organisational processes associated with the adequacy and appropriateness of information and assurance of providing it at the right time and in the correct, consistent and usable form.
2. Efficiency, which relates to the provision of information through the optimal (most productive and economical) use of resources.
3. Confidentiality, which relates to the sensitiveness of information to unauthorised disclosure.
4. Integrity, which relates to the accuracy and completeness of the information and its appropriateness in relation to organisational values and expectations.
5. Availability, which relates to the readiness to provide information on demand to business processes now and in the future.
6. Compliance, which relates to meeting the requirements of laws, regulations and contractual agreements binding for an organisation (both externally and internally).
7. Reliability, which relates to providing proper information to the management, enabling it to properly perform its fiduciary and governance duties.

As mentioned above, simultaneous and equal fulfilment of all attributes defined does not guarantee the quality of information. Besides, some of them are in opposition to each other (e.g., availability and confidentiality). Moreover, information needs, related to various information systems, vary. For example, in the case of a Website promoting the products of a given company, the availability of information is the most important feature, and in the case of a financial system – it is reliability, integrity and compliance. However, when considering the quality of information in the context of organisational management, thus in the context of a decision making process, it is commonly believed that the most important feature of information is usefulness, stressing its praxeological dimension (Kolbusz, 1993). Therefore, it seems that we can argue that effectiveness and efficiency are the most desired features of information in the list proposed by COBIT. An argument for this proposition is, for example, provided in the summary presented in Table 1.3.

[12] The COBIT model is discussed in more detail in Section 1.6.5.

Table 1.3 Summary of the number of primary and secondary instances of the various information criteria set out in COBIT 4.1 model processes

Information criterion	Primary	Secondary	Irrelevant	TOTAL
Effectiveness	25	4	5	34
Efficiency	23	6	5	34
Confidentiality	2	6	26	34
Integrity	6	11	17	34
Availability	4	13	17	34
Compliance	1	10	23	34
Reliability	2	14	18	34
TOTAL	63	64	111	238

Source: Own work

The purpose of each of the 34 processes defined in the COBIT model is to satisfy the relevant information criteria. This means that, for each process, criterion or criteria are defined, to be met in the first place, ones to be met in the second place and the criteria whose meeting is not significant at all (238 cases in total). As we can see, the primary criteria to be met most often are effectiveness and efficiency, because they highlight the two levels of pragmatic action. Effectiveness refers to taking appropriate action, and efficiency to doing this the right way. These criteria are indicated by T. Kotarbiński as the most important factors of effective action (1975, p. 457)[13]. Therefore, it seems that these two criteria are decisive of the quality of information in terms of its suitability to support the organisation management process. Other criteria are the means of ensuring them and are in a way the cost of their implementation.

1.4.2. The definition of an information system

Another key concept that requires more attention is an *information system*. This concept, combining the two terms, *information* and *system* is not only their simple combination. Therefore, the aim of further discussion will be to identify the elements of this synergy. The basic element pointing to it is the context in which the information system is embedded. As pointed out by E. Kolbusz (2004), the term *information system* was probably first used by N. Churchill, C. Kriebel and A. Stedry. These authors show the relationship between the information system and the organisation. The relationship

[13] Although Kotarbiński calls the second attribute thriftiness, however, he defines it as *cost-effectiveness* – the antithesis of waste which exactly corresponds to the definition of efficiency presented earlier.

between the organisation and the information system is also pointed out by R. Stamper (1973), who distinguished three levels of an information system: informal, formal and technical.

As shown earlier, teleology and holism are basic attributes of a system. In the further analysis of the concept, consisting of reviewing selected definitions of an information system, these two attributes will be included in a special way.

Another definition of an information system has been proposed by W.A. Bocchino (1975, p. 17) as follows:

> *A management information system arises from the need of the management for accurate, relevant and useful data, in order to plan, analyse and control the work of the company in a way that optimises its development. A management information system accomplishes this task by ensuring the introduction of processing and transmission of data and through a feedback network that enables managers to respond to current and future changes within the company and its environment.*

According to this definition, the purpose of an information system is to provide good quality data (accurate, timely and useful) that will be used to implement the management functions (planning, analysis, control). While this definition clearly emphasises the teleological dimension, it basically says nothing about the holistic aspect. It only refers to functions that fulfil the goal (input, processing and transmission of data) and points to its systemic origins (feedback network, environment).

On the other hand, W. Steinmüller (1977, p. 20) proposed the definition accenting the holistic dimension and indicating that the information system is a social model, consisting of combinations of the following components: (1) data, (2) processing programs, (3) technical means, (4) people who supply, process and use information, (5) the organisational structure of the system, and (6) the relationships of the system with the environment. It should be noted that this definition, due to the presence of man as a component, does not allow to classify the information system to the second group in the topology of systems described in Section 2.1, the designed artificial systems (artefacts). This definition places an information system in the last typology group in paragraph 2.1, i.e. the category of human activity systems.

The reservation made above, indicating the classification of the information system in the category of human activity systems is inasmuch significant as there is also a term ”computer-based information system”. This concept, although defined in various ways, usually focuses on its technical (computer) implementation, unfortunately, often forgetting about the most important ingredient – man. For example, Z. Bubnicki (1993, p. 76) defines an computer-based information system as:

> *... a system whose components perform the following functions: obtaining (mining, acquisition) of information, its transfer, collection, processing and receiving– where at least basic functions are performed by computers. The basic functions are processing and collection of information.*

It should be noted that the concept of a *computer-based information system* corresponds to the lowest level of the IS in the abovementioned distinction set out by R. Stamper. R. Ackoff (1967) cautioned against the dangers of too much emphasis on the role of computers as tools supporting the management process. He pointed to the risks of delegating organisational problems to solutions solely technical in character. Furthermore, definitions using the term *computer-based* are already somewhat out of date, even if we narrow our focus to technical solutions only. Along with the ongoing phenomenon of digital convergence, devices have appeared that have a much broader functionality than traditional computers, such as portable wireless devices (PDAs) and mobile phones. So in this sense, it is better to use the concept of an automated or technological component.

With few exceptions, in the early years of the development of the field of information systems, this term was understood in a broad context, rather than limiting it to the technical layer only. Only the rapid development of computer technology (microcomputer revolution), which began in the 80s, caused a detrimental transfer of emphasis from organisational to technical issues. This does not mean that the technical aspects are not relevant – they play a key role in the present phase of evolution of civilisation, but the emphasis on the division into information and computer elements is not always desirable. The innovative role of information technology is the focus of the next section of this chapter.

As authors presenting the infological approach, B. Sundgren and G. Steneskog (2003, p. 23) believe that a computer-based information system is only a special tool that helps humans in information processing. Many times the term information system in the sense that it is *computer-based* in fact refers to a data processing system because, as mentioned before, in the infological approach, information and an information system exist only in human minds.

Z. Pawlak (1983, p. 16) provides the formal definition of an information system. According to this definition, an information system is a group of four elements:

$$S = \langle X, A, V, \rho \rangle \tag{29}$$

where: X is a finite set of objects; A - a finite set of attributes; ; $V = \bigcup_{a \in A} V_a$; V_a – domain of the attribute a ; ρ – *the entire function,* $\rho : X \times A \rightarrow V$; where $\rho(x, a) \in V_a$ for each $x \in X$ and $a \in A$.

In the comments to the proposed definition, the author points to its narrowing to the field of information retrieval systems. The information system is a finite set of information about the objects of one type (e.g. people, books, or machines) and can be represented as an array whose columns are attributes and rows are attribute values for each object. It is worth noting that the definition proposed by Z. Pawlak is entirely independent from the technical means needed to implement the system itself.

J. Kisielnicki and H. Sroka (2007, p. 18) define an information system as:

A multilevel structure, which allows the user of this system to transform specific input information into the desired output information by means of appropriate procedures and models.

Specifying the holistic aspect, the authors show that the elements of an information system of any organisation comprise the set of elements:

$$IS=\{P, I, T, O, M, R\} \tag{30}$$

Where: IS means the information system, P – the set of subjects who are the users of the system, I – information resources, T – the set of tools used in collection, transmission, processing, storage and distribution of information; O – the set of system solutions, M – the set of meta-information; and R – relationships between the sets. Using the above equation, they define the concept of a *computer-based system* as a set, which relates to computer hardware (p. 20). According to them, a computer-based system is:

A separate part of the information system, which has been computerised from the standpoint of adopted goals.

The authors point out that the distinction discussed above is not deliberate, because in this age, all information systems are computerised, the problem is only the scope and specificity.

Similarly, Kuraś M. (2009) draws attention to the purposelessness of this distinction and draws attention to the danger of the isolated use of the term *computer-based IS*. Looking at an organisation from the perspective of computer-based IS restrict the functionality of created systems to the data layer only, which in consequence emphasises inappropriate criteria for determining the system purpose. Implementation of information systems, often simply understood as implementing standard off-shelf software, creates the need to adapt the organisational structure to the system and not vice versa. Kuraś notes that only a holistic perception of the information system allows for meeting the competitive challenges by applying innovative technical solutions.

B. Sundgren and G. Steneskog (2003, p. 22) define an information system as:

> *An organisation's total information system is made up from the mental concepts and frameworks of the participants in the organisation, the data passed to and between the participants (processed along the road), and the resulting individual perceptions and understandings of the situations leading to individual actions. These actions are expected to lead to the fulfilment of shared goals.*

This understanding of the system indicates the following components of an information system (p. 22): (1) Common understanding of goals; (2) Good communication; (3) Common culture, language and coding conventions; (3) Compatible frames of reference; (4) Common data and metadata. This definition, similarly to the definition by W. Steinmüller, emphasises the holistic aspects, stressing the importance of a man as a key component of the system and also drawing attention to the teleological aspect, but without defining it precisely.

W.A. Bochino's definition, despite its clearly teleological nature (providing data for the implementation of management functions), does not indicate what is the purpose of the organisation itself, leaving it to its specificity. In fact, organisations may be very different. However, the majority of them are businesses, that is, profit-oriented entities. We could of course argue that this narrowing excludes all other organisations. However, it seems that the adoption of such a position is justified, because the previously determined context of the field of information systems indicates that it is precisely this type of organisations which are within its main scope of interest. In the case of other types of organisations, whose character is not specified i.e., all kinds of institutions, foundations, and associations, we must be satisfied with a more general description of organizational objectives. Therefore, according to the discussion in the previous section, it seems that the objective of IS is to provide efficient information cost-effectively. However, further clarification of the goal of an organisation, where an information system is used, allows avoiding adopting too vague definitions. The effects of adopting vague definitions has already been pointed out in earlier sections of this chapter, but if we assume that it is appropriate, in accordance with an earlier suggestion, to limit the discussion to commercial companies, in this case the information system will contribute to the general objective of the company – to generate profit in the long term. The definition, which clarifies the purpose of an information system, taking into account the above claim, is proposed by M. Kuraś (1994):

A management information system is to provide decision makers with the necessary information, which will ensure that the organisation can adapt to changing conditions (environment), keeping in mind the main objective: to gain a sustainable competitive advantage.

A broader discussion on teleological and holistic aspects of information system, the values carried by its technical support and the management of the IT aspect at an organisation will be included in subsequent sections of this chapter.

1.4.3. The information system as a complex system

Each organisation is a complex system. A complex system can be defined as (Gomółka, 2000):

An open system[14], whose integral component is man bringing about a conscious and deliberate action of this whole, which is separated from the environment.

This definition emphasises the social dimension of the information system. The complexity of the information system also stems from the fact that it consists of other complex subsystems. The basic division taking into account the role of an information system within an organisation is shown in Figure Fig. 1.7.

A management subsystem includes all functions related to decision making, setting of objectives, direction and scope of activity (note that three levels of management are distinguished within an organisation: operational, tactical, and strategic). The task of the executive subsystem is the daily fulfilment of tasks set by the management subsystem. An information system in this model acts as a link between the management subsystem and the executive subsystem. This is a simplification of the tasks of an information system, because its effect is also apparent in the areas of both subsystems. Communication and sharing of information resources is essential on both the executive and management level.

[14] An open system is a system which interacts with the environment (Gomółka, 2000, p 12).

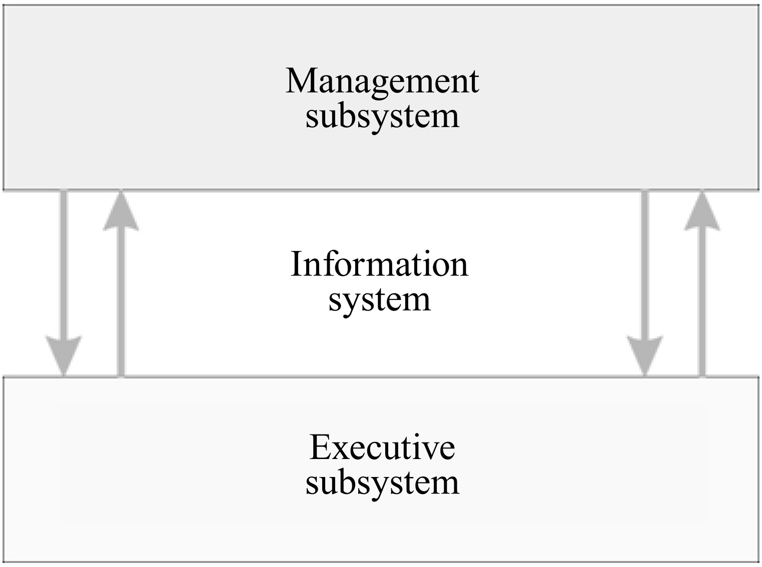

Fig. 1.7 The role of the IS as a subsystem of the organisation
Source: Own work

A more complex model of an organisation is the 7S model (Fig. 1.8), which was created in the international consulting company *McKinsey & Company* (Peters and Waterman, 1984). The main elements of the model are:

1. Shared values – the rules of conduct, the values connecting people who make up the organisation.
2. Strategy – the way of achieving the goals adopted by the organisation.
3. Structure – formal dependencies between the components of the organisation (including the formal organisational structure).
4. Systems – the method of implementing specific activities, processes (e.g. the system of production, distribution, compensation, as well as the information system).
5. Staffing – the method of adoption of new employees, requirements and specialities, jobs and skills of existing employees.
6. Skills – related to the organisation as a whole as well as presented by individual members of the organisation.
7. Style – the behaviour of the organisation's members in their mutual contact and in particular the style of management.

The name of the model comes from the names of individual components. This model provides a valuable basis for discussion about an organisation, because it takes into account both “hard” components – strategy, structure, systems, and “soft” ones – staffing, skills, style and shared values. The shared values are a kind of link and foundation for the formation of other components, because all elements of the organisation are conditioned by the principles underpinning the organisation. The essence of the smooth functioning of the organisation is the mutual compatibility of all its parts, both hard and soft, and it is much easier to modify the hard components. Changing the soft components requires a lot of time and extensive managerial and interpersonal skills. In particular, this applies to the common values that are directly related to the organisational culture. All organisation elements must be subordinated to the vision and mission, and they must support achieving of strategic objectives which the entire organisation pursues. This model also emphasises the social context of the organisation and its information system.

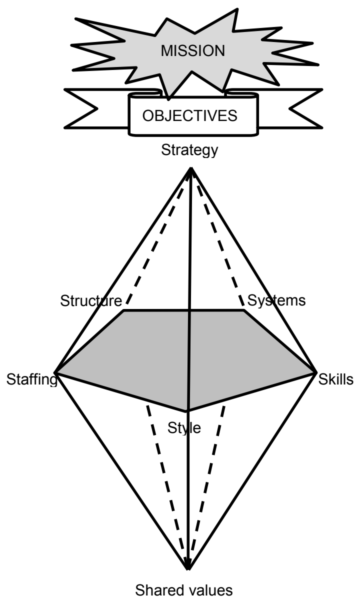

Fig. 1.8 The 7S model

Source: (Kuraś and Zając, 1995) based on (Peters and Waterman, 1984)

Information is the basic component/factor in the decision making. The quality of the decisions made depends on the quality of information. The personality traits of the decision maker also play a very important role in the decision making. We can talk about the willingness of taking risks by people who are ready to make a decision with a lack (deficiency) of information, while others use all available data sources trying to minimise this risk. However, decision making is always associated with uncertainty (risk) resulting from the existence of the information gap (Fig. 1.3), already mentioned in Section 1.2. This gap is related to the possibility of clarifying the problem situation and the availability of information in the time designated for making the decision. The decision maker never has complete information on a specific problem. Decision making in this context is always associated with risks. However, we can control this risk using information of an appropriate quality.

The quality of information[15] results from such features as (Kemball-Cook, 1973, Kuraś, 1981, O'Shaughnessy, 1975):

- Purposefulness – information must serve someone and something, there must be a rationale for collecting and using information.
- Reliability – concerns the veracity of both the source of information and its contents.
- Timeliness – information must relate to the decision making period and be delivered in a timely manner.
- Completeness – information cannot be random, it must take into account the decision-making context.
- Versatility – information should present the decision-making situation from many different points of view.
- Appropriate accuracy – it should be neither too detailed nor too general.
- Legitimate financial outlay – the use of information has to bring benefits in the amount that at least covers the expenditures for its acquisition.

In order to examine the quality of information, first we need access to data based on which information is created. For the data to be available at the time of decision making, it must first be collected i.e., the information needs of decision makers and the organisation as a whole must be anticipated. Information needs have several dimensions that can be categorised as follows (Zając, 2004):

- Universal – in some ways obvious, commonly known by the employees, even if they were not included in the software (such as date, location, company name, its address).
- Basic – essential in the conduct of activities, without which the company would not be able to function at all (e.g. company's offer, pricing, product range, etc.), this should also include mandatory data required by law.

[15] As mentioned in Section 1.3.2, information is not data, but an idea, image, abstraction – a significant component of knowledge, which the human being has, wants to and can use to take action (a decision). Data are an important factor allowing the decision maker to create information through the understanding and assimilation (internalisation) as a result of thinking processes (mental, conceptual). The required features of information presented are related equally to information itself as to data, but their assessment is entirely subjective, as it relates to the abstract creation, which is information.

- Extended – affecting the increase of effectiveness and efficiency of the organisation, market position or differentiation and extension of the offer, meaning e.g. potential customers, competitor's offer, possible sources of supply, etc.
- Potentially useful – related to the comprehensive tracking of new trends in the industry, political and technological changes in a broader context than just the business operations of the company. They provide the greatest potential for gaining competitive advantage and have the greatest potential to shape the development of the company and gain leadership.

There may be a risk, however, associated with the collection of excessive amounts of data, from which it is difficult to generate potentially relevant information, thus the creation of the IS must be subordinated to the actual information needs of the organisation. At the same time, when designing a new solution within the IS, we cannot be confined to the analysis of existing needs, but we must anticipate the future needs of the organisation. The starting point for such an analysis should be the vision and mission of the organisation, its objectives and strategy.

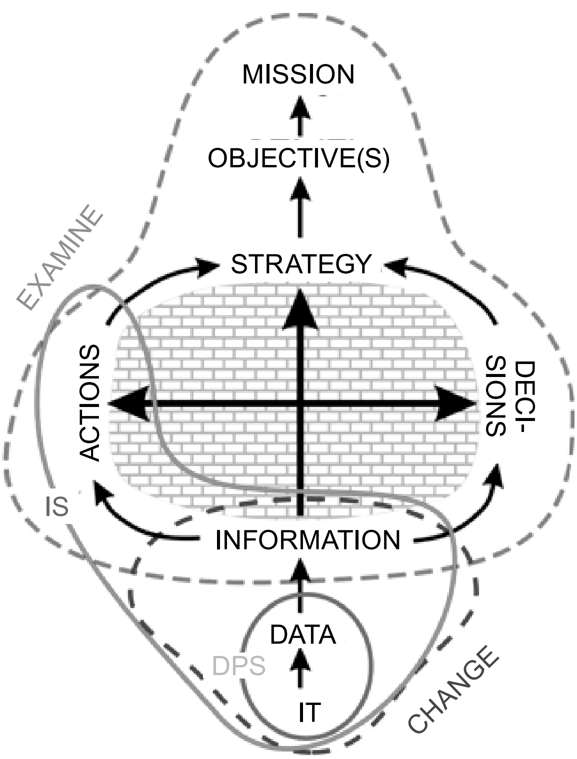

Fig. 1.9 Information model of an organisation
Source: (Kuraś, Zając, 2010)

Fig. 1.9 shows the organisational context of the operation and development of the IS. When making changes in the data processing system (DPS, which can be identified as a computer system), we must be aware that the information system is not limited to data and technical measures. It is an integral part of the IS, which in turn is an important component of the organisation.

Any interference in the IT area will affect the acquired data, and thus the information that shape decisions and actions. Decisions and actions taken will determine the fulfilment of the strategy and achieving organisational objectives.

When designing a new IS, we must not only remember about the basic functions (collection, gathering, storage, processing, retrieval, transmission, broadcast and presentation of data), but above all about its organisational context. Especially with regard to the strategic dimension of IS, it must consider not only the specificity of the organisation, but also be adapted to the knowledge, skills and needs of decision makers. Various kinds of systems are used to transform data into information, information into knowledge and (to some extent) knowledge into wisdom.

1.4.4. A typology of management information systems

In the initial period of the application of IT in business (in the 50s of the twentieth century), software was specialist and distributed in character. Usually, there were specialised applications designed to support specific areas of activity (such as accounting, payroll, inventory, etc.). Also, the data were gathered in a manner separated from other applications. This resulted in the need for duplication of data in some areas, which sometimes led to the emergence of errors associated with the existence of contradictory data (those manually entered in several different areas were not necessarily identical). In addition, decision-making capabilities, based on different data formats, presented in various forms, have been considerably limited. The need to link data to decision making at various levels of management was observed. In this way, in the 60s of the twentieth century, the first computer-based management information systems appeared[16] (Ackoff, 1967).

Transaction processing systems (TPS) are responsible for gathering operational data, recorded during the current activities of the organisation, at the executive level. Their task is to record all basic (detailed) data which can be later used to create any statements, reports, summaries, etc.. Transaction

[16] Although the subject of the typology of information systems is quite common in the literature in the field of IS (e.g. Kisielnicki and Sroka, 1999; Laudon and Laudon, 2002, Rainer and Cegielski, 2011, Zawiła - Niedzwiecki *et al.*, 2010), there are some differences in the method and scope of organisation of the existing types of IS.

processing systems are the basis for the creation and use of any other management-support systems, since they all derive from the data collected in them (Fig. 1.10). These data are collected in different areas of business activity (in figure Fig. 1.10 they are shown as the main functions of the organisation), regardless of whether the transaction processing systems are isolated from each other (independent), or integrated. In the case of connecting data on the executive level (transaction processing systems), we can talk about horizontal integration, which is best reflected in the ERP (Enterprise Resource Planning) systems discussed further in Chapter 3.

Office automation systems (OAS), which include tools such as word processors, spreadsheets, database management systems, presentation tools, graphical tools, electronic mail, or applications for scheduling (so-called *organizers*), are used at all levels of management and in the administration area of an organisation.

Expert systems (ES) may be potentially useful at all levels of management, but in practice they are not very common due to their complex structure and high cost. These are systems based on artificial intelligence (AI), typically used in narrow specialist fields. Their essential feature is the use of expert knowledge, which is accumulated in the knowledge base. The user can use expert advice without direct contact with experts through the expert systems.

The task of management information systems (MIS)[17] is to support the resolution of operational and tactical problems at lower levels of management. The action of a MIS is to search, compile, present, combine and analyse data from various sources (primarily from transaction processing systems of the organisation, but it is possible to extract data from external sources) to provide information for making routine decisions. An important feature of this system is keeping track of changes and deviations of the results from the assumed level on an ongoing basis. Management information systems support the broad range of tasks in the organisation, including analysis and decision making. Examples of the application of management information systems include controlling current production or short-term (operational) planning.

[17] It should be noted that management information systems (MIS) have a double meaning in English. In a narrower sense, they are one of the types of information systems, i.e. systems for informing the managers, and in a broader meaning, they specify all types of management information systems and the whole field dedicated to the creation and operation of information systems in management.

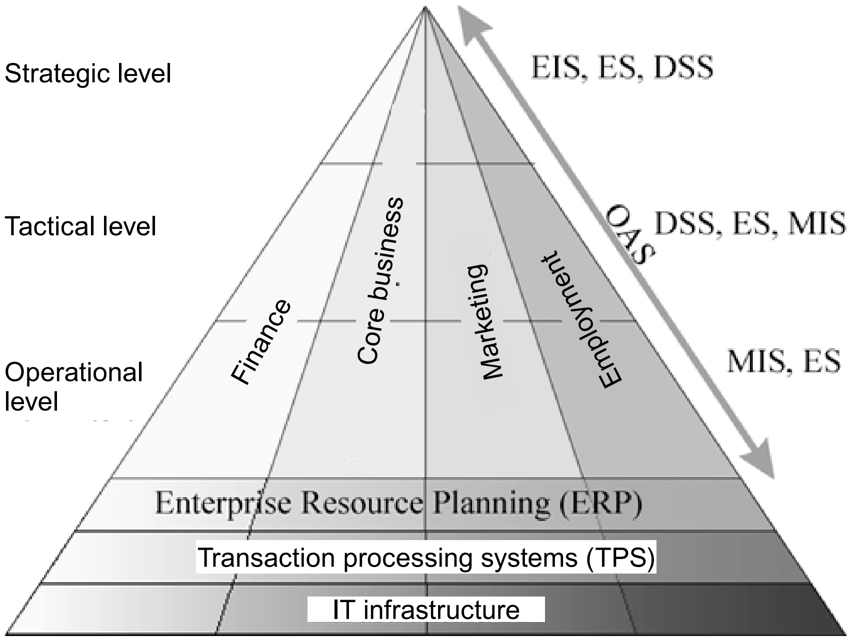

Fig. 1.10 Types of IS within the organisation
Source: Own work based on (Rainer and Cegielski, 2011)

Decision support systems (DSS) are interactive systems which support strategic and tactical planning. Because it is possible to change the functions of these systems (they are subject to continuous development), this allows for adjusting the system to changing conditions. Just as management information systems, they are based on transaction processing systems as the data source. In contrast to the MIS, where we can only get the data related to the past, DSS enable to obtain answers to the question *What if ...? by using a database and a base of methods (models).* They simulate possible scenarios, thus facilitating the making of certain decisions.

The purpose of executive information systems (EIS) is to provide information necessary to make decisions at the highest level of management. They are equipped with a data protection mechanism that allows for restricting access to strategic information. Their operation is based on access to information of global organisations, their selection and integration. Their primary function is to narrow/select the necessary data and their appropriate processing, so that on the one hand the information is reliable and on the other – not redundant. Remember that at the highest level of management, a comprehensive overview of the situation of the company is needed, based on all available data sources, but on the other hand, a detailed analysis of all data is not possible because of their abundance and the limited time from the decision. Therefore, the task of EIS is to provide rapid aggregate information with the simultaneous ability to analyse the causes of problematic situations through the access to more detailed information.

Currently, the development of IT applications in the area of decision support introduces a lot of tools to help rapid diagnosis of the problem. These tools are collectively referred to as business intelligence (BI). They allow for the selection and such presentation of the data, so that decisions at all levels of management can be made as soon as possible. Mechanisms related to BI include the data mining systems already mentioned in Section 1.2, OLAP (OnLine Analytical Processing) and digital dashboards.

1.5. INFORMATION TECHNOLOGY

1.5.1. Technological progress and organisational progress

The impact of technology on the nature and type of work, as well as on the economic development became in the last century the focus of research in both sociology and economics. This subject was especially taken into account in the discussion in the field of institutional economics born in the U.S. in the 20s, which combines the two above-mentioned scientific fields (Rutherford, 2001, p. 174). The representatives of the institutional school of economics, sometimes called the "old institutional economics" (T. Veblen, W. Mitchell and J.R. Commons), analysed the development of world economy in the context of the institutions created by men. They have brought attention to the behavioural aspect of human action, but what is particularly interesting in the context of the discussion included here, they also focused on the role of modern technologies and their impact on behaviour, the way of living and thinking. The discussion on the role of technological progress was continued on the basis of new institutional economics and economic history by D.C. North. In his work (1974), North indicates that the technological progress is widely believed to be the source of the success of the Western World. However, as far as it can be considered a necessary condition of civilisation development, it is not a sufficient condition. Actually, the significant change in the methods of management diversifying the world economy in favour of the western economy took place during the Industrial Revolution, and its main achievements have been associated with (North, 1974, p. 7):

- Machines substituted man's hands;
- New sources of energy were developed;
- Man was able to transform and utilize matter in a new and revolutionary way.

If, however, the economic development depended solely on technical development, considering the fact that technical solutions are generally available at low cost and are easy to copy, the global economy would show no diversification. Therefore, D.C. North believes (1974, p. 8) that the progress of civilisation is a direct derivative of three factors:

1. Development of human capital that is able to adapt, modify, and develop technology;
2. Presence of physical conditions allowing to deploy and implement selected technology;
3. Existence of efficient economic organization.

The above discussion suggests that along with the technical factor, the human capital and efficient organisational skills are needed in obtaining the higher level of development, which allows for the formulation of a thesis that the civilisation development is not so much the result of technical development, as of organisational development. A more detailed discussion on the relationship between information technology and the institutional environment is included in the work (Grabowski, 2007; Grabowski and Dymek, 2010).

1.5.2. The role of information technology in modern business organisations

Information technology (IT) is a key asset of modern business organisations. It consists of the whole of technical tools and methods necessary for the implementation of information systems, namely:

- Hardware (computers, communication equipment and other)
- System and application software
- Data stored on electronic media
- Data communications infrastructure that allows integration of hardware, software and data

The dynamic development of the elements related to the broadly defined communication in the last decade, resulting in, among other things, the development of the Internet and mobile phones has meant that sometimes it also bears the name of information and communication technology (ICT).

The term *information technology* began to appear more frequently in the literature in the field of MIS with the advent of the revolution of microcomputers, i.e. starting from the 80s. Previously, the technological dimension did not stand out in any particular way and the information system was treated as a whole. This does not mean that information technology was not used previously in the tool layer of information systems. This concept has always been present in one form or another in discussions in the field of IS. Focus on the technical aspects was one of the most important factors (Leavitt and Whisler, 1958), which from the beginning defined the essence of the field of information systems. However, at the beginning of the 80s, the concept of *information technology* began to be used more and more often as a separate term, considered in different contexts. The major topics in the 80s and early 90s include primarily the role of IT in planning, particularly in strategic planning where it may be a factor providing a competitive advantage (Rackoff and Wiseman, 1985; Bakos and Treacy, 1986; Ward, 1986, 1988; Boynton and Zmud, 1987; Doll and Vonderembse, 1987; Ives and Vitale, 1988; Johnston and Carrico, 1988; Copeland and McKenney, 1988; Tavakolian, 1989, Jarvenpaa and Ives, 1990, Lindsey et al., 1990) and other topics such as the impact on the organisational culture and management (Olson, 1982; Foster and Flynn, 1984; Zmud, 1984; Sutherland and Morieux, 1988) or an attempt to determine the relation of IT investments to effectiveness (Weill, 1992) and value (Dos Santos et al., 1993) of a company.

One of the first comprehensive monographs discussing the role of information technology in business context was the work of M.S. Scott Morton (1991). It proposes an organisational model called MIT90 (Fig. 1.11). The model name comes from the fact that it was developed by a team working at the Sloan School of Management at MIT (Massachusetts Institute of Technology) in the U.S. This model was the first in history to take into account technology as one of the factors influencing the market position of a company. It indicates the need to ensure a balance between structure, business processes, strategy, organisational culture, practices and processes in the area of human activities and the technological dimension, which largely consists of information technology. The technology is included among those factors that can and should directly interact with the strategy of the organisation, because it has the potential to help in providing a competitive advantage. The organisation also must remain in balance with the socio-economic and technical environment. Compliance with the technical environment consists in adapting internal solutions to the capability of cooperation with other businesses and consumers. In this regard, the development of private and public computer networks and the need for the development and adoption of universally recognised standards has been suggested.

As for the role of the information technology itself and its impact on changing the nature of the work performed in businesses, its important role has been pointed to in three aspects:

1. Extensive communication capabilities, including multimedia
2. Capabilities of high-volume access to databases in order to obtain the appropriate information
3. Universal availability of equipment allowing the use of the above-mentioned functions in a manner similar to that in which people communicate with each other

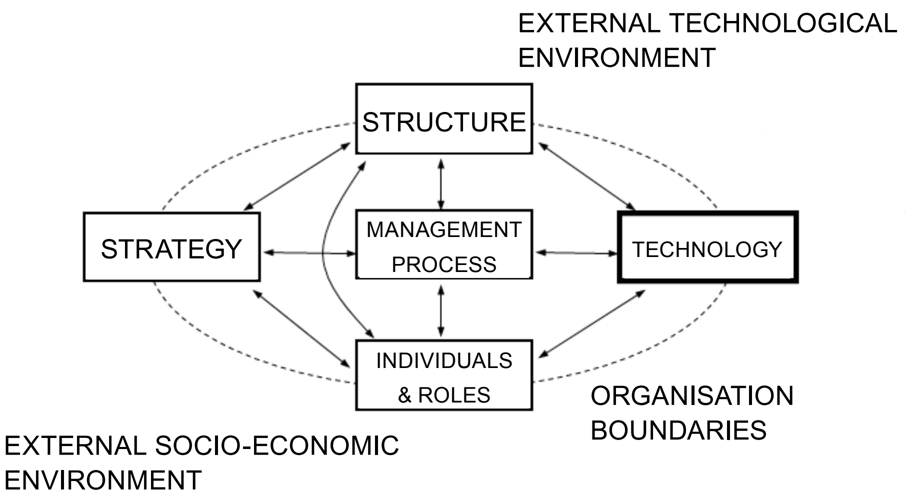

Fig. 1.11 MIT90 model
Source: (Scott Morton, 1991)

In the further work of the group organised around the Sloan School of Management, published in a subsequent monograph, edited by T.J. Allen and M.S. Scott Morton (1994), an important guiding principle was formulated, according to which (p. 3):

> *No longer will information technology simply be overlaid onto an existing business; it will now be used to restructure the enterprise.*

This thesis was a result of the research carried out by the authors among companies who were than the market leaders. These results suggested that innovative use of information technology has led to a dominant market position. Emphasising the role of information technology as a key factor in organisational change is one of the main postulates of the approach known as business process reengineering (Hammer and Champy, 1993).

1.5.3. The concept of the waves of innovation

The importance of information technology in the creation of organisational value is well illustrated by the concept of the waves of innovation (Fig. 1.12) in (Primozic et al., 1991). The authors define five ways, according to which IT was used for the past fifty years to automate and support business processes. Ways to use IT to support business, i.e. the waves of innovation, are organised according to the time of their historical appearance. The benefits that each of the waves brings were examined by the authors from two perspectives: from the perspective of the functional use and the management focus. The years when the specified wave was considered innovative are marked on the abscissa.

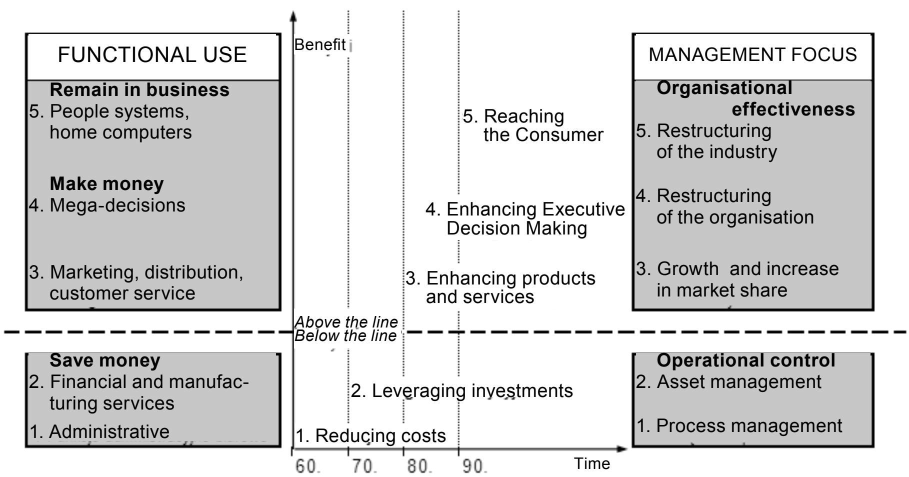

Fig. 1.12 The waves of innovation
Source: (Primozic et al., 1991)

The first wave, which took place in the 60s, consisted mainly of automation of tasks previously performed manually, allowing for significant cost savings. The automation of financial and accounting functions is an example of this.

The second wave of innovation began in the 70s. Its main objective was to optimise the use of corporate resources. Examples of the applications of this wave are the MRP / MRP II (Manufacturing Resource Planning) systems, more widely discussed in the third chapter of this textbook. The systems created in the first and second wave of innovation, aimed at automating and supporting administrative, financial, and production functions, are focused internally and can be implemented at any time, focus on reducing costs and therefore are under the dashed line.

Three subsequent waves of innovation that are above this line are designed to support income generating activities. The third wave began in the early 80s. This was when the strategic dimension of IT was recognised for first time. Applications relating to it were designed to increase market share and improve external organisational functions such as marketing, distribution and customer service. Fourth wave coincided with the second half of the 80s.

Applications of the fourth wave include systems supporting decision-making processes at various levels of organisations – now often described as business intelligence systems.

The fifth wave of innovation, which was just beginning at the time of the cited publications, has gained particular significance following the substantial development of the Internet. It was very noticeable in the second half of the 90s and is now the dominant phenomenon. Understanding its essence has become the key to defining the organisational strategies of the present day. For the first time in history, it allows for the implementation of the concept of the "pull" economy on a global scale. The concept of a pull economy, as opposed to a "push" one, is that it is the customer and not a given economic organisation, who triggers, parameterises, and even models the economic processes.

1.5.4. The strategic significance of IT

The enthusiasm of the 90s led to a situation where many perceived the information technology as a factor, which in itself is a source of competitive advantage. It was not until the turn of the century when the crisis of Internet start-up companies (dot-coms) clearly revealed that just using IT cannot equal the organisational change.

One of the most quoted publications criticising reckless investments in IT is the article by N. G. Carr (2003). He compares IT to other, as he calls them, infrastructural technologies such as the steam engine, railways and electricity in his article. In conclusion, he claims that since the availability of these goods is not currently a source of competitive advantage, the availability of IT does not have such a significance any more. Due to the fact that IT solutions are widely available, it is easy to copy and implement them. What's more, reckless investments in IT are potentially very high risk. Therefore, organisations should focus primarily on the following activities: (1) reducing costs, (2) refraining from innovation, and (3) paying attention primarily to weaknesses and not the opportunities that IT brings.

The article mentioned has met with widespread criticism (Stewart et al., 2003; Goliński, 2004) and, although it is difficult to deny the N. G. Carr is right in the issue of increasing vigilance and discipline in spending related to IT, it is difficult to agree with most of his theses. For it cannot be argued that IT does not matter, when an argument is brought that 50% of corporate spending of U.S. companies is investments in IT. It is also difficult to agree with the thesis of the disappearance of the strategic importance of IT, as it has a great potential in making organisational improvements. Moreover, the strategic dimension of IT related not so much to the availability of the solutions itself, but above all to the innovative potential of an organisation, has never yet been so important in history because of the increasing degree of sophistication and complexity of business processes. It is the nature and innovative competence of an organisation that decides whether a solution brings benefits or losses.

The fundamental mistake made by N. G. Carr was equating IT with the other industrial technologies which, although they have some similarities, in fact vary considerably. The main difference is that in contrast to the industrial technologies, IT has an intellectual character. K. F. Curley and P. J. Pyburn (1982) believe that the main distinguishing feature of intellectual technologies is that their function is not apparent in the physical design. In general, we can point to similar elements that make up a particular artefact in all IT solutions, i.e. input devices, output devices, memory, CPU, and the fact that these elements perform mathematical and logical operations. What distinguishes individual cases is their logical structure. It is defined by logic programming, which essentially does not change its physical structure. An artefact with a programmed logical structure in each type of application implements a different purpose. Moreover, this goal has not and is not generally known at the time of construction of the artefact. We could therefore say that information technology as opposed to industrial technology is general in nature and thus multi-purpose. In contrast, industrial technology arose as a result of a solution to a specific problem, and thus, has always been targeted at specific applications. Due to these factors, the ability to apply information technology requires the organisational learning process (Nolan 1973; Nolan and Gibson, 1974; Nolan 1979).

Another feature that distinguishes IT from industrial technology, as pointed out by M. J. Earl (2003), is its ambiguity expressed in the risks associated with its use. These risks can be both a cause of failure and success, and it is this aspect of IT that is closely associated with its potential for innovation. Speaking of the uncertainty of IT, M. J. Earl points to its three dimensions: (1) uncertainty about the nature of solutions, (2) uncertainty of verification of a given solution, and (3) uncertainty about the impact on the organisation and the scope of implementation.

Uncertainty about the nature of solutions results directly from the fact that IT is a general-purpose technology. The use of a particular technology is not always consistent with its initial vision. We could bring up the example of the World Wide Web, which surpassed the imagination of its creator, T. B. Lee, and mobile text message (SMS) technology, which to some extent has cannibalised mobile voice communication. Uncertainty about the nature of solutions implies the following managerial problems: (1) It is necessary to replace the linear model of strategic planning with experimental models, which will allow much more efficient alignment of the business strategy with the potential which IT gives. (2) This task cannot be left solely in the hands of technologists, but it is necessary to involve business managers and users. (3) Measurement of the effectiveness of new solutions should not only focus on traditional short-term measures, but should also take into account factors such as organisational learning.

Uncertainty as to the verification of a given solution relates to the question whether this solution works in practice. History may show many cases of solutions that simply have not worked, though initially were predicted to succeed. One example is the French Minitel (Cats-Baril and Jelassi, 1994), which was quickly replaced by free Internet. Uncertainty about the verification of a given solution implies the following managerial problems: (1) If the solution is characterised by a high risk, then on the other hand it is also an opportunity. (2) In view of the potential risks, tactics to eliminate them should be used, which often means employment of specialists working with the suppliers of given technologies. (3) Metrics associated with the process of assimilation of new technology are an important tool for measuring efficiency in the case of uncertainty of the return.

Finally, uncertainty about the impact on the organisation and scope of implementation is related to the question whether the stakeholders of the organisation will adopt the new solution or whether it will allow for operation in a particular organisational context. Also, the following managerial implications exist in this case: (1) Whether a particular technology fits into the context of the organisation. (2) If the uncertainty in question is high, it is necessary to involve stakeholders affected by this process. (3) Performance metrics include both economic and social elements.

The first two of these uncertainties are closely linked to innovativeness of solutions that are designed to achieve competitive advantage. However, the advantage gained by the synergy arising from linking technology with the organisational conditions may not always be easy to copy – the examples being such companies as *Google*, *Amazon.com*, *Allegro.pl*, *CISCO* or *Dell*. On the other hand, failing to face new challenges may be the cause of the collapse of companies, even with a very established market position, such as *Digital Equipment*, for example.

Uncertainty about the impact on the organisation and the scope of implementation is not related to innovation, and generally does not provide a competitive advantage. ERP systems discussed in Chapter 3 are examples of solutions in this group. Their essence is generally known and they are commonly verified in practice, but the incorrect implementation of this kind of technology resulting from the wrong determination of the organisational context, erroneous parametrisation and adaptation or failure to take account of behavioural aspects results in a lack of return on investment, increasing the costs and decreased organisation performance relative to the state before the implementation, which in turn would weaken the competitive position of the company.

1.5.5. IT productivity paradox

Unfortunately, the lack of awareness of the conditions discussed above makes it possible to currently observe the phenomenon that investments in the IT area usually end in failure. This phenomenon is called the IT productivity paradox. ITGI (2006) reports the following data:

- According to the Gartner Group report from 2002[18], about 20% of all IT spending was wasted, which annually amount to around $600 billion.
- Research conducted by IBM in 2004[19] among Fortune 500 companies indicated that approximately 40% of all IT spending brought no returns.
- According to a study of the Standish Group from 2004[20], only 29% of IT projects succeeded, while the rest either did not produce the intended results, or has been terminated.

Despite a fairly long period of use and numerous studies on the economic theory and practice, it appears that information technology and its role in contemporary organisations is still not yet sufficiently well understood. Some confusion arises without a doubt from its dissimilarity in relation to other industrial technologies. The second part of the confusion is related to the overestimation of its role as a primary condition for organisational change, which is equivalent to the emphasis only on the second of the previously presented postulates by D. C. North. However, most IT specialists still consider information technology as a decisive factor for competitiveness of modern companies, if the organisational aspects of innovation associated with its use are properly appreciated, and because the probability of failure is high, as shown by the above figures, the more one shall make every effort to properly control

[18] Gartner Group, The Elusive Business Value of IT, August 2002.

[19] Watters D., IBM Strategy and Change, survey of Fortune 1000 CIOs, Presented to SHARE in New York, 17 August, 2004.

[20] The Standish Group International, Third Quarter CHAOS Report, 2004.

IT issues. We will discuss a new paradigm for managing the IT area – IT governance – in the next, which is the last paragraph of this chapter.

1.6. IT GOVERNANCE

1.6.1. The conditions necessitating the creation of a comprehensive IT governance model

In the last quarter century, the role of IT as an integral component of information systems has significantly increased. In the 50s, IT solutions supported only few critical organisational functions. The 60s and 70s were characterised by the development of isolated systems, whose main role was played by mainframe computers. Starting from the 80s, interconnected mini- and microcomputers became increasingly important, and together with telecommunications elements they formed the IT infrastructure. In the 90s, thanks to the Internet, the IT infrastructure began to take on global dimensions. P. Weill et al. (2002) show that while speaking of the IT infrastructure at present, it is necessary to refer to its three layers: public, general corporate, and departmental.

This trend was reflected in the amounts the companies were spending on IT. In the article cited earlier, N. G. Carr (2003) quotes the data from the Federal Bureau of Economic Analysis of the U.S. Department of Commerce, which show that the capital expenditures attributable to IT rose in the last half century in a significant pace. While in 1965 they accounted for only 5% of corporate spending by U.S. firms, in 1980 – it was 15%, in 1990 – 30%, and in 2000 – already about 50%. As a result, IT has become one of the most important and costly organisational resources.

The turn of the century showed the importance of two groups of problems that significantly affected the perception of IT as a resource of contemporary organisations. These are:

1. The low effectiveness and efficiency of investments in IT, and
2. The need for control of the IT environment in terms of security, business continuity and compliance with the internal and external arrangements.

The problem of low efficiency of investment in IT infrastructure stems directly from the IT productivity paradox, and because it has already been outlined in the previous section, only the second of the above problems will be described here.

The dynamic growth of the Internet has caused an unprecedented increase in the availability of information systems, which in turn meant that they became much more susceptible to the risks of security breach. The terrorist attack on the World Trade Center in 2001 or the 2004 tsunami highlighted the scale of the problem associated with the need to ensure business continuity. The U.S. corporate market scandals on an unprecedented scale, related to counterfeiting of financial documents[21] shown the need for fundamental legislative changes in order to increase investor confidence in financial markets.

The need to ensure control of the IT environment was reflected in the fact that for the first time in history, legal solutions were applied on such a large-scale. The most important step in this direction was the adoption of the U.S. law, named after the names of its authors as the Sarbanes-Oxley Act (Sarbanes and Oxley, 2002). This Act relates directly to the U.S. corporate market, but in an indirect way also applies to operators around the world, including in Poland.[22] One of the major objectives, described in Section 404, is to increase the effectiveness of internal control. Despite the fact that Section 404 is only half a page, the fulfilment of its requirements turns out to be a very costly task. The Act itself does not explicitly mention the need to extend the internal control system to the IT area, but because information systems are currently the source of financial data, the extension of the internal control to the resources and processes that use information technology has become a necessity. The exact scope of work related to the implementation of the Sarbanes-Oxley Act, including that relating to the IT layer, is determined in the applicable documents by The Public Company Accounting Oversight Board (PCAOB)[23].

The above conditions cause that the need for a comprehensive approach to the management of IT area has been noticed – it is now no longer possible to perceive IT as a secondary factor, and information systems can no longer be treated in an isolated way. In traditional terms, the planning of technical support was part of the strategic planning of the company. But now, due to increasing competition and shorter planning horizons, clearly defined action plans are replaced by "emerging strategies", consisting in the rapid use of the arising opportunities. B. C. McNurlin and R. H. Sprague (2002), include the following as the factors hindering long-term IT planning:

- Difficulties in ensuring compliance of business objectives with system objectives.
- Emergence of rapid technological changes.

[21] The most publicity being gained by *Enron* and *WorldCom*.

[22] The Act also applies to all subsidiaries of U.S. companies listed on U.S. stock exchanges, and entities registered and/or operating outside the U.S. but listed on U.S. stock exchanges.

[23] http://www.pcaob.org/

- Internet era, which causes that organisations require a portfolio of projects rather than isolated projects.
- Difficulties in financing the IT infrastructure.
- Interdisciplinary character of projects expressed by the need to connect the functional responsibilities of various functional verticals in the organisation.
- Other factors, which include the problem of coexistence of top-down and bottom-up project management methodologies and the need to reconcile the project with the character of a radical change with those consisting of a gradual improvement.

The same authors point to the emergence of new trends that cause change in the traditional IT functions, among which they mention:

- Emergence of distributed systems
- Takeover of system analysis tasks by business departments of the organisation
- High availability of off-shelf software packages
- IT services outsourcing

The subject of the discussion in this section is IT governance, an area that is a prerequisite for the existence of a new paradigm in the management of information technology (Grabowski, 2008a). It defines the decision-making areas and mechanisms and promotes the tools helpful in the systems approach to the problems described above (Grabowski, 2008b).

1.6.2. The origins of the concept

The concept of IT governance is the point of focus in both academic and business discussions, in particular in regard to auditing. The leading academic centre is the *Center for Information Systems Research*[24] (CISR), operating at the *Sloan School of Management, MIT*, USA. The leading business institution is *IT Governance Institute*, operating under the *Information Systems Audit and Control Society*[25] (ISACA). It should be noted, however, that this division is rather conventional, since the work of both these groups are complementary. CISR pursuing its research activities, works very closely with the business environment, while the activities of the other organisation are often attended by the employees of universities (e.g. the *University of Antwerp Management School*). Because of its area of interest, the first of these groups emphasises the scientific aspects, while the second focuses on application aspects. However, this distinction is guided by the nature of publications of these two groups.

[24] http://mitsloan.mit.edu/cisr/

[25] http://www.isaca.org/

The article by R. W. Zmud (1984) was one of the first academic works on the subject of IT governance. Although the author did not use the same term, he indicated the need for changes in the organisation of information systems departments and its relation to business divisions due to the growing strategic importance of information technology. The changes he described were supposed to affect a number of issues such as the choice of appropriate organisational structures and processes and implementation of appropriate management and incentive systems to ensure adequate interpersonal relationships.

The term *information technology governance* was first used by L. Loch and N. Venkatraman (1992). The authors, describing the process of outsourcing of IT services, which was novel at that time in the American economy, showed that the organisation of the IT infrastructure is slowly moving away from purely hierarchical and market mechanisms in the direction of hybrid relations integrating external suppliers. This article for a certain period of time remained in a certain isolation from mainstream publications, and the term *IT governance* or *IS governance* has gained some popularity only in the late 90s. Such terms as the *IS governance solutions* (Brown, 1997) or the *IT governance arrangements* (Sambamurthy and Zmud, 1999) appeared in academic publications at that time.

The mainstream academic work focused on trying to find the determinants of decisions correlated with the centralisation/decentralisation of certain functions related to the adaptation of IT infrastructure to business requirements. It was noted that these relationships result in the formation of three types of structures in the economic practice: centralised, decentralised, and federated. Publications in the 90s indicated the following regularities related to companies with complex multi-department structures (Brown, 1997, p. 70):

- Organisations used a centralised model, if the aim was to ensure operational efficiency and synergy of enterprise systems, and a decentralised model in order to ensure effective departmental control and the flexibility and responsiveness to customer needs.
- Organisations used a centralised structure in the case of responsibilities relating to the management of technical infrastructure, and decentralised solutions were generally used for the responsibilities associated with the use of specific solutions, including systems development.
- In some cases, it was reasonable to use the newly created federated model, which allowed combining the best features of both the centralised and the decentralised model.
- At the same time, research has shown that there is no single universal model for all companies. The best model for the given company depends upon its organisational context, and that these dependencies are complex (Sambamurthy and Zmud, 1999).

A. E. Brown and G. G. Grant (2005) have made an extensive study of literature containing over 200 articles in this group. The authors divided them into two trends: (1) the forms of IT governance, and (2) IT governance contingency analysis. The authors pointed to the monograph by P. Weill and J. W. Ross (2004), as a summary of these research trends.

IT governance is also being developed in the business practice, but its pragmatic aspect is emphasised much stronger than in the academic community. In the business circles, in addition to issue of efficiency and effectiveness of IT solutions, which is the main area of current academic discussion, the issue of compliance with external (institutional) requirements and the issue of security and business continuity are also extensively discussed. Given the application context, publications focused on business emphasise the role of tools used for the effective implementation of IT governance. The most important role in this area is played by the already mentioned COBIT model, as well as other models and standards, such as ITIL (IT Infrastructure Library)[26] (Cartlidge et al., 2007) and the Information Security Standard ISO 27001. These models are complementary in nature, but it is believed that the COBIT model is the most holistic in nature. Business publications develop and promote these models as well as other techniques, stressing the role of measuring the degree of compliance of the IT infrastructure with business goals, such as, for example, the balanced scorecard (Kaplan and Norton, 2001; Van Grembergen et al., 2004; Grabowski, 2009). This group of publication includes works by: W. Van Grembergen (2002), W. Van Grembergen and S. De Haes (2004), E. Guldentops (2004), S. De Haes and Van Grembergen (2008).

1.6.3. The definition of IT governance

There are many definitions of the concept of IT governance in the literature on the subject. The most frequently quoted definitions include the one proposed by P. Weill and J. Ross (2004, p. 2), according to which IT governance (2004, p. 2) is:

Specifying the decision rights and accountability framework to encourage desirable behaviour in the use of IT

Another frequently quoted definition is the one proposed by ITGI (2007, p. 8):

IT governance is the responsibility of executives and the board of directors and consists of the leadership, organizational structures, and

[26] http://www.itil-officialsite.com/home/home.asp

processes that ensure that the enterprise's IT sustains and extends the organization's strategies and objectives.

Similar to the above definition is the one proposed by Van Grembergen (2002) ([after:] Van Grembergen and de Haes, 2004, p. 5). According to this definition:

IT governance is the organizational capacity exercised by the board, executive management, and IT management to control the formulation and implementation of IT strategy and in this way ensuring the fusion of business and IT..

All of the above definitions indicate the structural nature of IT governance. It seems that the first one, originating from the contingency theory[27], places a stronger emphasis on behavioural aspects and is less deterministic. The second and third definitions more clearly emphasise the institutional and application aspects by setting IT processes in the context of organisational procedures and providing them with a legal framework.

The concept of IT governance must be distinguished from the concept of IT management (Peterson, 2004). The relations between IT governance and IT management are illustrated in Table 1.4.

Table 1.4 The relations between IT governance and IT management

Criterion	IT management	IT governance
Relations	Internal	External
Time horizon	Present	Future
Orientation	Process	Structure
Nature	Dynamic	Static

Source: Own work

IT management focuses on current activities and generally concerns internal problems of an organisation. This term is most commonly associated with

[27] Contingency theories belong to the group of behavioural theories. According to them, there is no single, ideal way of managing an organisation. On the contrary, certain factors make the management style and methods effective in some cases, are ineffective in others. According to the contingency theories, there are many factors that determine the organisational success, in other words, organisational success depends (is contingent) on many factors, such as the management style, environment, organisational subsystems, organisational tasks, task forces, etc. The most famous contingency theories include: In the context of leadership: Fiedler's contingency theory and the Hersey-Blanchard situational leadership theory; in the context of decision making: The Vroom-Yetton theory of decision-making participation, also called the normative decision theory; in the context of organisational rules: Smith's contingency rules theory.

the control of projects designed to implement specific solutions. It is therefore dynamic in nature and process-oriented, thus relates to the problems at the operational level. On the other hand, IT governance is concerned with long-term actions and takes into account the environment in which the organisation operates. The purpose of IT governance, which is structurally oriented and has a static character, is to create a framework for action, conditions and rules for the efficient governance of the IT area. This structure constitutes the meta-rules for participants in the process of IT governance. IT governance focuses on the issues on the tactical and strategic levels.

1.6.4. Systemic dimension of IT governance

It was emphasised in this Section 1.2 that the most important features of systems methodology is holism and teleology. It seems that it is due to the above clearly defined attributes that the concept of IT governance resembles systems methodology (Grabowski, 2008b).

According to the definitions quoted in the previous section, the goal of IT governance is:

- Encouraging appropriate behaviour in the use of IT
- Provision of IT skills in maintaining and expanding organisational objectives and strategies
- Control of the process of formulating and implementing the IT strategy, thereby ensuring the alignment of the IT area with business activities

The teleological dimension of IT governance is mentioned by Weill and Ross (2004), who perceive it as one of the integral parts of corporate governance. Referring to the definition of corporate governance adopted by the OECD (Nestor, 2001) they suggest that IT governance should take into account two aspects: behavioural and normative. The list of IT governance objectives presented above clearly highlights these aspects.

P. Weill and J. Broadbent (1998) detailed the benefits of applying the principles of IT governance in the implementation of the company's mission. Research conducted by the authors has shown that companies using IT governance rules get the return on investment in IT of up to 40% higher than their competitors. These companies are trying to look for value in using IT in one or more of the following ways:

- Clearly defining the business strategy and determine the role of IT in its implementation
- Measuring and managing expenditures and returns on IT
- Allocating responsibility for the organisational changes needed to reap the benefits of IT

- Drawing conclusions and learning from each single implementation in order to better share and re-use IT resources

The above list may be considered as four key areas of action for governing the IT area, clearly specifying the purpose of a technically-supported organisational change. It details the rather general concept of value resulting from the use of IT and provides a framework for appropriate behaviours associated with its use. In this sense, it specifies the teleological dimension of the IT governance issues. The list is sorted in descending order of *added value* of a given area. It attributes the highest value to the support of strategy by IT, but also does not deny the value of such areas as the focus on measuring inputs and outcomes, delegation of rights in carrying out organisational change, or only on learning from own mistakes and rational use of previously created IT resources.

The holistic dimension of IT governance is reflected in the complexity of the concept. It is necessary here to take many aspects, problems and areas of action into account. As mentioned above, IT governance is one of the integral parts of corporate governance. Like other parts of corporate governance, it requires the design of adequate rules associated with long-term management. These rules should be designed and implemented in such a way as to support the policies of corporate governance in the broad sense of this concept, taking into account both behavioural and institutional aspects. As a result, IT governance is an ordered set of decision-making areas and bodies, structures, processes and mechanisms aimed to adapt the IT area to business requirements – it will be called the IT governance structure later in this paper.

1.6.5. IT governance structure

Since IT governance is a relatively new concept, the number of available models describing the structure of IT governance is small. It seems that two of them reached a relatively high level of maturity. These are (1) the model belonging to the academic approach, described in the monograph quoted earlier (Weill and Ross, 2004), named the CISR model here, after the name of the centre where it was developed, and (2) the COBIT model belonging to the business approach.

1.6.5.1. The CISR model

The CISR model (Weill and Ross, 2004) examines IT governance at three levels: objective, subjective and instrumental. Summary of the elements for each level is shown in Table 1.5. The objective dimension of IT governance is composed of five decision-making areas relating to IT. Weill and Ross specify them as:

1. IT principles, i.e. the rules relating to the role played by IT in the organisation. They are defined at the highest strategic level of the organisation.
2. IT architecture, i.e. the rules associated with the logical data model, applications and infrastructure. This decision-making area is a set of rules allowing for the clarification of the requirements for standardisation and integration of IT solutions in both technical and business terms.
3. IT infrastructure – defines the shape and technical implementation of IT services and solutions. It specifies the requirements for equipment, processing model, software and data communications solutions. The main difference between IT architecture and IT infrastructure is that if the former is a design of IT solutions, the second is its implementation.
4. Business applications needs – stem directly from the organisation's business needs and determine how the use of specific IT solutions creates value in the company.
5. IT investment and prioritisation – selects the projects that support the mission of an organisation to the highest extent.

The second dimension for defining IT governance is to establish a decision-making body within a specific area of decision making. As the importance of IT governance increases with the size of the organisation, the issue of a decision-making body goes beyond single persons and usually affects multiple organisational departments. The first rather natural axis is a division into the IT department and business departments. In order to determine the interrelationship between the IT department and the business sphere, six archetypes are adopted.

When discussing various models of decision-making bodies, these authors distinguish between decision-making centres/persons and those who provide the information needed to make a decision. The different models of the subjective dimension of IT governance in the context of their particular archetype will be discussed below:

1. Business monarchy – decisions related to the IT area are made by managers of business departments of the organisation.
2. IT monarchy – decisions related with the IT area are taken by the IT department. This type of decision model is used in organisations in which the requirements for infrastructural cohesion are important.
3. Feudal – based on the sovereign-vassal type relations. It often boils down to a situation in which a given business unit has its own autonomy and defines the shape of the IT area for its needs on independently. The IT department in this case provides IT services, in return being financed by the unit it serves.

4. Federal – aimed at balancing the responsibilities of various units of the organisation. These units usually include the central units, business departments or process lines and the IT department. There is a fairly widespread view that it is most difficult IT governance archetype to implement, because of the difficulty in developing a consistent criterion optimising the objective of the organisation.
5. Duopoly – this archetype is used when the IT department is separated from the organisation as an independent business unit and provides services to other units on the basis of mutual agreements.
6. Anarchy – the name of this archetype speaks for itself. In this case, decisions are made at random and ad hoc. It is also difficult to determine which of the entities makes them.

Among the tools allowing for implementing the instrumental dimension of IT governance, Weill and Ross mention: (1) decision-making structures, (2) alignment processes, and (3) communication approaches. The following tools were identified on the basis of research in 256 corporations from 23 countries (Weill and Ross, 2004, p. 87).

In terms of decision-making structures, P. Weill and J. W. Ross mention, among others, the following collective bodies: the Board or a committee composed of top management, process teams together with representatives of IT and managers of the business/IT relationship. The authors suggest that certain mechanisms can be used in certain archetypes, but not in others.

Decision-making structures are the first step in effective implementation of IT governance. The next step is to use the specific matching processes, which include for example: management of IT projects, service level agreements (SLA) and formal tracking of business value of IT.

The last phase of implementation is associated with the proper transfer of information within the organisation about the role IT plays there, which certainly largely impacts the implementation of the principal objective of IT governance, which according to the definition adopted here, consists of encouraging appropriate behaviour in the use of IT among all employees of the organisation. It is to be ensured by the tools belonging to the group of communication approaches, such as announcements of the top management, a CIO office or IT governance as well as portals and intranets.

Table 1.5 IT governance structure according to the CISR model

Dimension	Decisions/archetypes/tools	Significance
Objective	IT principles	The role of IT in the organisation
	IT architecture	Standardisation and integration requirements
	IT infrastructure	Services shared and solutions implemented
	Business application needs	Business needs of the organisation
	Prioritisation of investments	Selection and funding of projects
Subjective	Business monarchy	IT decisions are determined by business departments
	IT monarchy	IT decisions are determined by the IT department
	Feudal	IT decisions are determined by the sovereign-vassal relationship
	Federal	Balancing the responsibilities
	Duopoly	Based on the relation business side – IT side
	Anarchy	No rules
Instrumental	Decision-making structures	Executives or senior management committee IT leadership committee comprising IT executives Process teams with IT members Business/IT relationship managers IT Council comprising of IT and business and IT executives Architecture committee Capital approval commitee
	Alignment processes	Tracking of IT projects and resources consumed Service level agreements Formally tracking business value of IT Chargeback agreements
	Communication approaches	Work with managers who don't follow the rules Senior management announcements Office of CIO or office of IT governance Web-based portals and Intranets for IT

Source: Adapted from Weill and Ross (2004)

1.6.5.2. The COBIT model

The COBIT model (Control Objectives for Information and Related Technology) has been developed since 1996 by the IT Governance Institute, an organisation working closely with the Information Systems Audit and Control Association – ISACA. This model helps in determining the business value that involves the use of IT and emphasised the compliance of the control objectives with legal regulations. The last of these postulates is reflected in the fact that it was recognised next to the of COSO standard (*Committee of Sponsoring Organisations of the Treadway Commission*), which is a widely recognised internal audit standard, as recommended to ensure compliance with the already mentioned Section 404 of the Sarbanes-Oxley Act.

Its main purpose is to define and measure control objectives related to the use of IT to support business processes. It was created as a set of good practices, which undoubtedly accounts for its application usefulness. The authors of this model include both university staff and practitioners from many corporations. Also in Poland, the COBIT model is a recognised standard and is widely used as a comprehensive model for auditing information systems and IT governance, and is a reference model for the proposed solutions (Kotulski et al., 2007).

In its structure, COBIT defines, among other things:

- Seven information[28] criteria: effectiveness, efficiency, confidentiality, integrity, availability, compatibility, and reliability.
- Four types of IT resources: applications, information, infrastructure, people.
- Thirty-four processes organised into four areas: Plan and Organise, Acquire and Implement, Deliver and Support, and Monitor and Evaluate. The composition of each process includes specific actions.

The factor binding individual components of the COBIT structure is a system of measures defined for each process. It consists of the following elements:

- Maturity models – used for comparisons and identification of directions for improvement.
- Indicators of objectives and measures of the implementation of objectives – COBIT distinguishes two groups in this group of indicators/measures:
    1. Lag indicators/Key goal indicators (KGI)
    2. Lead indicators/Key performance indicators (KPI)

Detailed studies on the implementation of IT governance based on the COBIT model can be found in the studies of E. Guldentops (2004) and ITGI (2007).

IT governance represents a new approach using methodology to address the issues of return on investment in IT infrastructure and to take into account

[28] The information criteria of the COBIT model are described in Section 1.4.1.

the problems of security and business continuity as well as to meet institutional and legal requirements. In the future, it may constitute a new paradigm for managing the IT area, but in the present era is still in the pre-paradigm phase (Grabowski, 2008a).

Over the last decade, the subject of IT governance has been developed both in the theoretical and practical dimensions. The publication by Weill and Ross (2004) has a fundamental significance to the realm of a theoretical discussion – the proposed cognitive model, which contains decision-making areas, archetypes, and tools should be subjected to a closer empirical verification – this need is indicated by a number of publications. The importance of IT governance for business practice consists in identifying problem areas of which IT governance is composed and proposing tools to assist in the implementation of specific mechanisms. In practical terms, particularly important becomes the trend associated with the COBIT model, related models and other techniques emphasising the role of measuring the degree of of business/IT alignment.

2

ANALYSIS AND DESIGN OF INFORMATION SYSTEMS

2.1. SOFT SYSTEMS METHODOLOGY

Despite the frequent use of the general systems theory, briefly discussed in Section 1.2, in the context of social sciences, this theory has many critics, among other reasons, due to its general and too universal character. One of the researchers criticising the general systems theory is P. Checkland (Checkland, 1993; Checkland and Holwell, 1998). For many years, P. Checkland has been working as a researcher and practitioner where he repeatedly used methods based on general systems theory. Unfortunately, these methods have proved ineffective in the case of complex systems in which the human factor plays an important role. P. Checkland drew attention to the fact that in the case of social problems in general, and of information systems in particular, it is difficult to assume that the phenomena studied in them are determined and repetitive in a similar manner as it is in physical reality, which has become the inspiration for the creation of the general systems theory.

In the case of an information system, we are dealing not that much with the physical reality (buildings, computers, software), but the social construction (Berger and Luckmann, 1983), which consists of projections of people regarding objectives and the role a given system plays in the organisation. As a result, Checkland has developed the soft systems methodology (SSM), which is an alternative on the one hand, and on the other a supplementation to the general systems theory (which P. Checkland called the hard systems methodology).

The soft systems methodology uses systems concepts taking into account the subjective and nondeterministic nature of social sciences. Formulating assumptions of the SSM, Checkland (1993, pp. 161-162) defined them as a set of general guidelines and not as a method or methodology, understood in the sense of the common meaning of the word *methodology*, where the action of an accurately described algorithm gives standardised results. Interestingly, he refers to the praxeological[29] understanding of the term *methodology here, quoting the eminent Polish philosopher, Prof. T. Kotarbiński.* Such understanding places SSM between philosophical (responding to the question *What?*) and pragmatic aspects (responding to the question *How?*). This means that the SSM includes

[29] Universal Dictionary of Polish Language PWN defines the term *praxeology* as: ‘<praxeo- (from Gr. práksis, prákseos ‘practice’ + -logy> philos. “general theory of efficient operation, the field of scientific research related with all purposeful human action” (http://usjp.pwn.pl/haslo.php?id=1865536). Prof. T. Kotarbiński is considered the precursor of Polish school of praxeology. One of the most compact monographs in the field is (Kotarbiński, 1975).

in the scope of its discussion both aspects of defining the analysed system and the tools using which it will be implemented. SSM puts a slightly different emphasis on the issue of systems methodology itself than GST does. In GST, the essence of the systems methodology is the assumption that reality consists of systems and the goal is to identify them. In SSM, the emphasis is on organising (systematising) the process of understanding reality in the form of an iterative learning process (Checkland and Holwell, 1998, p. 157).

The following assumptions are the basis of SSM (Checkland and Holwell, 1998, pp. 157-162):

- SSM is used for creating information systems aimed at solving poorly structured or unstructured decision problems[30].
- It is necessary to take into account and explain both fixed and variable components of the analysed social reality.
- The hard systems methodology is a special case of the soft systems methodology. It can be used for the analysis of fixed components. When dealing with both fixed and variable elements, it is necessary to use the soft systems methodology. The task of the hard systems methodology is to optimise a particular solution, while the task of the soft systems methodology is to interpret and participate in the learning process.
- SSM is based on the interpretive approach[31].

The central idea of the approach proposed by Checkland is the concept of a human activity system. Checkland derived this category from the following typology of systems (1993, pp. 110-111):

[30] In management theory, decision problems are divided into three groups. The first one includes structured problems, where the decision maker has to deal with a decision situation of a standard and/or repeatable character, for which both the objective and the decision-making options are clearly defined and techniques supporting the decision-making process are known. In opposition to this first group is the third group, consisting of unstructured problems, i.e. those in which we are dealing with a complex decision-making situation in which for example the decision-making options are not fully known or are not comparable and the objective is not measurable. The decision-making procedures are also unknown. Between these two types of decision problems lies the second group of semi-structured problems. In this case the decision situation is composed of well-structured elements (e.g. well-defined and comparable decision-making options) as well as unstructured ones (e.g. not fully measurable objective).

[31] The interpretive approach is a commonly used research approach in MIS, next to the positivist and critical approaches. It presupposes that reality is socially conditioned and therefore it is knowable only through interaction with people forming and using it intentionally (Sundgren and Steneskog, 2003). In the interpretive approach, reality is not knowable in an objective manner and can be at best interpreted by the researcher. The aim of interpretive study is an attempt to draw reasonable inferences in a subjective or intersubjective way (i.e. true for a group of people and not just the individual). Research results are only a snapshot of reality in certain circumstances and time, so it cannot be used to draw general conclusions.

Natural systems. They include all the systems observed in the universe that were not built by man but are the result of processes occurring in nature. Natural systems show the order of the universe. In particular, their hierarchical order plays an important role. Man as a biological system is also a natural system, but fundamentally different from other natural systems because it is the only system which creates systems of another kind. According to Checkland, natural systems are the only ones that are the result of evolution. However, in contrast to the reductionists, he believes that they are "irreducible wholes". Checkland understands irreducibility as significant statements that address natural systems as wholes (Checkland, 1993, p. 113).

These statements remain true even when considering their components and the relationships taking place between them.

Designed physical systems. These systems, also known as artefacts32, contain elements of natural systems and are the result of an intentional (teleological) human creative process. Their characteristic feature is that they exist in physical form and satisfy a certain need of their creator.

Designed abstract systems. Just as designed physical systems, these systems are the product of a deliberate human action. However, they have no physical form. They are a structured conscious product of the human mind. Examples of such systems are areas of knowledge, including mathematics, philosophy, or art, such as poetry, theatre. They can be preserved in the physical form, i.e. books, diagrams, or computer programs.

Human activity systems are the last of systems classes in the described typology. These systems bring together certain mutually interacting areas of human activity that may be considered as a whole. These systems are inherently complex. An example of such a system is the air transport system. It consists both of designed physical systems (planes, infrastructure), designed abstract systems (flight schedules, procedures, methods of optimising loading) and the people, without whose intervention these systems would not operate. A characteristic feature of human activity systems is the fact of self-awareness of its basic component – man. Human activity systems are characterised by much greater variability and indefiniteness than natural systems.

A human activity system, as already mentioned above, is the primary object of research of soft systems methodology. Its exact definition is as follows (Checkland, 1993, s.314):

---

[32] Universal Dictionary of Polish Language PWN defines the term *artefact* as: <Lat. arte factum 'artificially produced' "something which is the work of the human mind and human labour in contrast to the products of nature". (http://usjp.pwn.pl/haslo.php?id=1810562)

The human activity system is a notional purposive system which expresses some purposeful human activity, activity which could in principle be found in the real world. Such systems are notional in the sense that they are not descriptions of actual real-world activity (which is an exceptionally complex phenomenon) but are intellectual constructs; they are ideal types for use in a debate about possible changes which might be introduced into a real-world problem situation.

A human activity system is composed of two subsystems (Checkland and Holwell, 1998, pp. 159): (1) a teleological subsystem of activities and (2) a control subsystem ensuring the survival of the whole in changing circumstances.

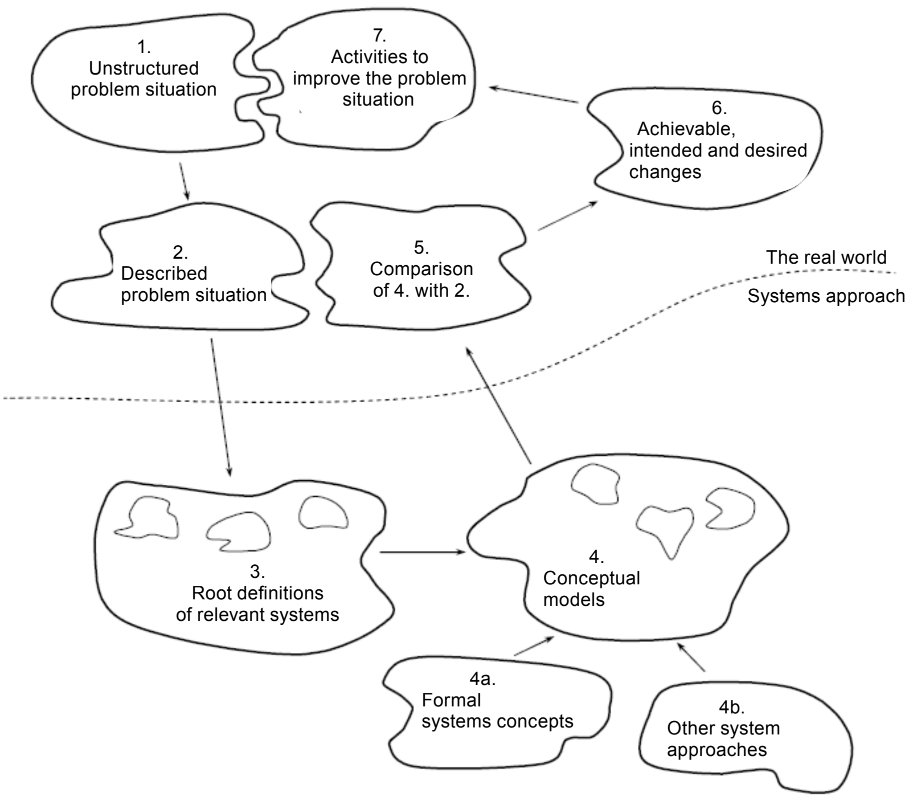

Fig. 2.13 Soft systems methodology
Source: (Checkland, 1993, p. 163)

The following describes the different phases of the soft systems methodology. The numbers of individual phases (Fig. 2.13) point to its sequential nature, whereas the start of the modelling process itself can take place at any stage. The final design of the system usually requires several iterations of the entire cycle.

Phases 1, 2, 5, 6 and 7 take place in the physical world and necessarily involve people who are the components of the problem situation. Description language used in these phases is the language understood by the people, who are components of the system of actions, i.e. it is a natural language supplemented by drawings and sketches. Phases 3 and 4 (4a, 4b) are conceptual in nature and are handled by experts in the field of system modelling (optionally they can involve people who are the components of the problem situation). In this case, the description language takes a more formal form, expressed through, among other things, a consistent application of system concepts.

Phases 1–2: Description of the problem situation. The first and second phase are to identify and describe the complex and unstructured problem situation. This is accomplished by obtaining the richest possible picture of the problem taking into account the largest possible number of points of view. When identifying the problem situation, it is necessary to distinguish between static (structural) and dynamic (process) elements. People involved in this phase are usually the actual or potential members of the analysed system of action. As a result, the description of the problem situation takes the form of a set of so-called rich pictures, containing all the essential elements and relations of the analysed system. Rich pictures are a collection of texts written in natural language, supplemented by sketches, diagrams, and other drawings, which in contrast to the data flow diagrams (DFD) or entity relationship diagrams (ERD) are less formal in nature. The product of phases 1 and 2 is a list of names (and descriptions) of systems corresponding to the problem situation (it answers the question: what? should be done). This list provides a starting point to determine changes in order to solve the problem (the answer to the question how? to do it), which is the subject of subsequent phases.

Phase 3: Root definitions of relevant systems. The problem situation described in the form of rich pictures is then subjected to a process of formalisation. The individual elements that define the purposeful activity are determined using system concepts, the so-called root definitions. Any purposeful activity can be interpreted in different ways by different stakeholders of the human activity system. The SSM system introduces the following system concepts defined by the acronym CATWOE. Individual words making up the term are:

- Customer – a beneficiary or victim of the system
- Actor – a person who performs specific roles in the system. The sum of the actions of all actors defines a transformation
- Transformation – the process of converting the input into the output

- Weltanschauung (or Worldview) – the context of the transformation carried out
- Owner – a person having authority over the system
- Environment – external components which need to be taken into account. These include organisational procedures and political, legal and ethical conditions

Root definitions are hypotheses identifying potential directions for future actions, which are both appropriate (relevant) and possible to implement from the point of view of the involved people and system analysts. A properly formulated root definition should be a concise description of reality drawn up from a certain point of view *(Weltanschauung)*. It should specify the system owner acting within certain limits of the environment, and a transformation performed by actors, which directly affects the beneficiaries (or victims) (customers) of the human activity system.

Phase 4: Determination and verification of conceptual models. This phase aims to determine the models for the system of activities, which will ensure the implementation of the transformations described in the root definitions. Root definitions define what the system is, and the conceptual models determined in this phase specify the actions that systems must perform. This description is a logical sequence of action and should abstract from its physical implementation in the real world. Phase 4 is the most formalised part of the SSM. It consists of two (sub)phases: 4a and 4b. The conceptual model is determined in phase 4a. An essential part of the conceptual model is a description of actions that the system must perform in a transition. Therefore, its description should use few verbs that most comprehensively capture the essence of the system of actions. The next step carried out in phase 4a is the formal verification of the conceptual model defined earlier. According to Checkland (1993, pp. 173–174), in order to verify a conceptual model, we can use a formal system model. According to the criteria of the formal system model, each conceptual system should meet all of the following criteria: (1) be goal-oriented, (2) allow measurement of the effectiveness of action, (3) contain the elements of the decision-making process, (4) contain components which have the characteristics of systems (subsystems), (5) its components must interact with one another, (6) be a part of a larger system, (7) have a formally defined environment, (8) have both physical and abstract resources, and (9) be characterised by continuity of action (ability to survive). Phase 4b contains all other system methods and techniques that can be used to define and verify the conceptual model, in particular the GST.

Phase 5: Comparison of conceptual models with reality. This phase is not entirely a comparison, because we are dealing with objects of various types: a formal structure resulting from phase 4, and an informal understanding of the problem described in phases 1 and 2. The essence of the comparison is to confront the model drawn up by the system analysts with an intuitive understanding of the problem situation by its participants. This comparison can be carried out in many ways, but the type of the proposed system decides about choosing a particular method. Checkland (1993, pp. 178–179) lists the four most popular ways to make the above comparison:

1. Structured questions. In this case, the formal models developed are the basis for a constructive discussion. This is the most common comparative method.
2. Reconstruction of the sequence of events from the past. It allows to verify the model in terms of the correctness of responses generated by the model to events that make up the problem situation.
3. General comparison. It is based on determining which features of the conceptual model significantly differ from the actual problem situation and why-
4. Applying the model to the problem situation. It involves creating a model of the problem situation, i.e. expressing it in a formal language, the closest possible to the model described in phase 4.

The above-mentioned comparative methods are the basis for discussion about the necessary changes.

Phase 6–7: Implementation of the feasible and desirable changes. The purpose of this phase is to discuss the possible and intentional changes in the analysed problem situation. As mentioned above, the change must be both intentional and possible. The fulfilment of this postulate usually results in a compromise between the goals of the system and limiting conditions determined by the environment and between the goals of the system contained in various formal models. Achieving this compromise often requires several iterations of the SSM process. In the case of the “hard systems methodology”, the result of this phase is system implementation, whereas in the case of the “soft systems methodology”, the system implementation is considered a special result of phase 6. In general, the result of phase 6 is carrying out a more moderate change. Checkland (1993, p. 180) distinguishes three types of changes that may occur individually or together. These are:

1. Structural changes. They concern static elements of the problem situation and are associated with changes in relationships between elements. They can consist in grouping or dividing the elements into subsystems.
2. Procedural changes. They are associated with dynamic elements. They consist of changes in the implementation of individual processes and other activities occurring in static structures.
3. Changes in attitude (behaviour). They concern the social elements of the system of actions – i.e. people. They include all categories of behaviour conditioning the success of changes carried out, such as changes of group

consciousness, expectations, readiness for change, and the evaluation of the behaviour of others.

The author points out that while the first two types of changes are relatively easy to implement, this last category is the most serious challenge. Taking the behavioural dimension into account is a crucial factor that distinguishes SSM from other systems methodologies.

The characteristics of the soft systems methodology described in this section, expressed in the consideration of behavioural and organisational aspects, as well as a peculiar lack of determinism in the quest to achieve the optimal solution, causes that this methodology has an advantage over the general systems theory in the analysis of social systems.

2.2. OVERVIEW OF THE SYSTEM LIFE CYCLE MODELS

Initially, when the developed software was relatively simple, and problems were structured, software development did not require special design methods. With time and the increasing complexity of software, defined methodologies were developed and used to develop software at first and then to design the broadly defined information systems. The waterfall model was the first model that has found widespread applications in the field of IS. In the waterfall approach of elaborating on various stages, the design tasks are gradually and progressively determined. The waterfall model assumes a sequentiality of the stages of building the solution, assuming that each stage begins when the previous one finishes. This model has been used for decades, organising the process of creating the system, but with the increasing scope and complexity of designs, its flaws began to be noticed, such as time-consuming design work and limited opportunities to make changes during the design work (a solution, once adopted, was only elaborated on in various stages). An organisation was required to precisely determine the strategy, because all the design work was based on it. Under the conditions in which the project lasted many months, creating a good solution required extensive knowledge and experience from the authors in order to anticipate future needs of the organisation, enabling it to adapt to market requirements. As a result of these flaws of the waterfall model, approaches aiming to eliminate them began to appear, such as the incremental model, or the spiral model.

The basic premise of the new approaches was an evolutionary achievement of a final result, assuming the possibility of modifications to the requirements and the repeatability of stages (iterativity). Examples of such approaches include the incremental model (Fig. 2.14) and the spiral model (Fig. 2.15).

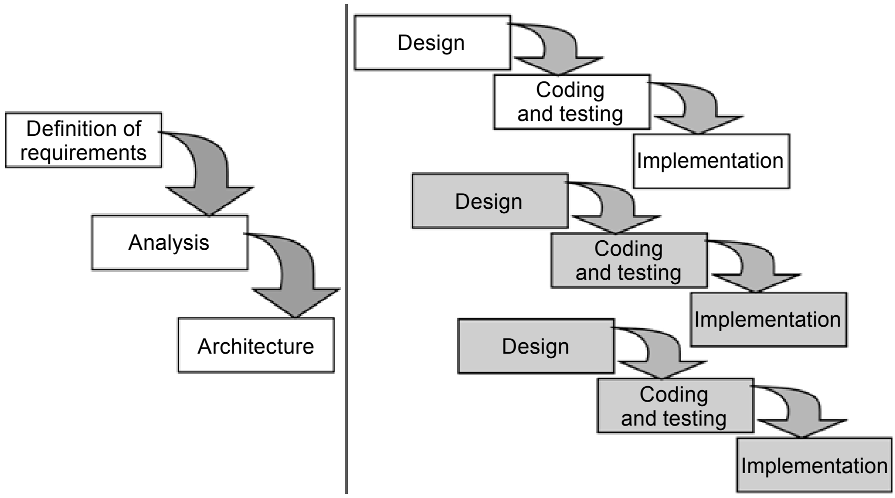

Fig. 2.14 Incremental model
Source: (Szyjewski, 2004, p. 37)

The incremental model starts with the overall business plan, defining the main (strategic) requirements and adopting initial assumptions about the system architecture. Further areas of the system are created in stages (increments), whose number depends on the size and complexity. Some sub-designs can be executed in parallel, others may be postponed. This method allows for a phased achievement of the final effect, which can and should be used to better match the changing demands of the environment. It is also possible to expand the system in the future. However, of crucial importance are the first phases of the project (vision of the whole system), which ensure the integrity of the solution. Please note that the entire creation process is spread out in time, and each additional module is composed into the already existing solution. Therefore it is important to clearly define the final effect. The incremental model, although based on the waterfall model, due to the division of work, can be used in large projects where labour intensity and necessary outlays are too great to bear in a short time. An additional benefit of this approach is the possibility of quite early use of the parts of the system created without having to wait for the implementation of the IS as a whole.

The spiral model proposed by Bohem (1986), takes into account the risks associated with the project in addition to the evolutionary achievement of the final IS form. Same as in the incremental model, the basic life cycle phases of the system are repeated in the subsequent cycles of creating the solution. Apart from the risk analysis, this model assumes early involvement of users in its design (each prototype is evaluated and subsequent requirements are determined by the users). This type of approach allows for the verification needs and at the same time learning the software being implemented.

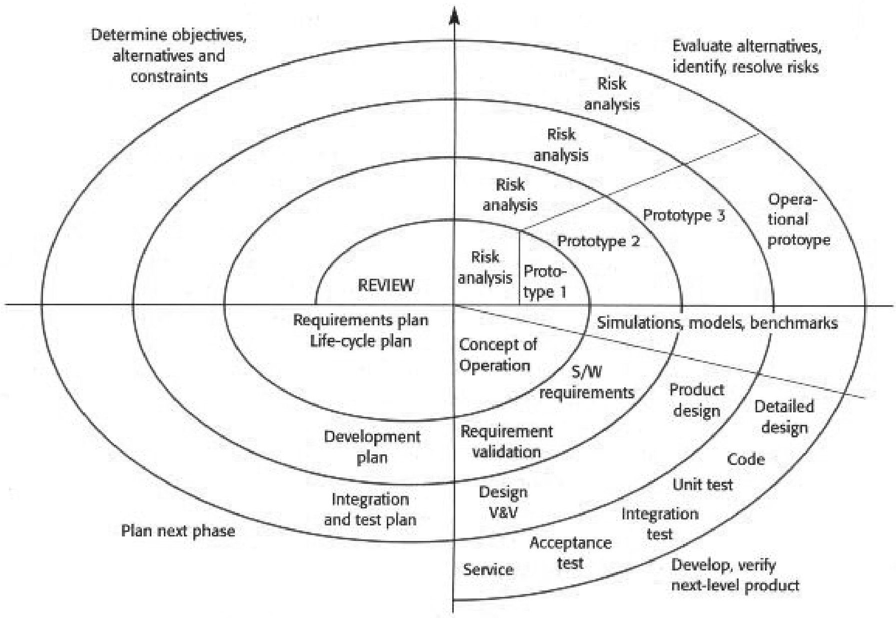

Fig. 2.15 Spiral model
Source: Own work based on (Bohem, 1986)

There are many other approaches and methods for the design of IS (JAD, RAD, prototyping, agile methodologies), but certain steps are common in most approaches, and the differences are due to their scope and repetition (iteration).

2.2.1. The project lie cycle

The phases of the project of information system development are often called the project development life cycle. Certain stages in creating the project are (or at least should be) included in all the design approaches and models. These stages are:

- Feasibility study (sometimes called the planning phase, or requirements specification)
- Analysis of needs, problems, existing IS

- Logical design
- Detailed/technical/physical design
- Implementation, software development
- Testing and trial operation

The whole design work starts with the concept of the system resulting from the needs or problems which the organisation observed in the area of information efficiency. This stage consists of the definition of the problem, which directly precedes the feasibility study phase, which can be defined as a preparation for a decision as to the sense of the project, taking into account the circumstances (limitations) of the project (see Kroenke, 1992). Work at this stage includes:

- Defining objectives
- Defining the limitations of the project
- Determining the feasibility
- Conclusions and recommendations

The concept of the new system should be reflected in well-defined objectives, whereas it is worth considering the three types of objectives: the organisation's objectives, the objectives of the information system and the objectives of the project. When proposing the SMART method, P. Drucker stated that the objectives should be:

- Specific
- Measurable
- Achievable
- Realistic
- Time-related

One more requirement should be added to the foregoing: the objectives should also be ambitious.

When starting the project, we should primarily aim at recognising the objectives of the organisation as a whole, which is the responsibility of the board, and the role of IS analysts is to reach the actual objectives guiding the organisation. The objectives of the information system are subordinated to the objectives of the organisation and should indicate the role of IS in achieving them. It is, in some way, a way to indicate the expectations from the future level of effectiveness of the information system. Project objectives are subordinate to both the objectives of the IS and objectives of the organisation and should indicate the parameters according to which the system will be evaluated after its implementation.

After defining the objectives of the project, we should look at the conditions under which the project will be built and in which the IS will operate in the future. A good starting point is to define the constraints of the project, which should be investigated, taking into account the following categories:

- Financial
- Time-related
- Organisational
- Technical
- Legal
- Related to knowledge, skills and attitudes of people
- Political (not always present)

In the area of financial limitations, first of all the allowable project budget which the organisation can afford is taken into account. We should remember about the future operating costs of the system, such as IT staff salaries, license fees, costs of maintenance, energy and consumables, subscription fees, etc. Typically, the choice which is more expensive in the design phase and less expensive in the operation phase is preferred.

Considering the time constraints, we must take into account the time in which a working solution must be created, and possible disruption in the process of creating the system (e.g. seasonality of the organisation's operations, other projects or implementation planned, planned changes in employment structure, etc.)

Organisational constraints relate primarily to the policy and character of the activities of the organisation, industry or market requirements. These can be such elements as the production system used, the use of standards, agreements with partners, the statute of the organisation, employment policy, or distribution system. Some of the conditions will affect the possible solutions in the process of creating the project, other will affect the organisation of the design process itself.

Technical conditions of the project are ones that will affect the selection of possible IT tools for the use in the project. They may result from the previously used solutions (e.g. format of the data, which will be imported to the new system), location (whether company is based in one location/building, or for example, many remote offices, and maybe so-called mobile employees) or the type of activity (such as the need to install computers in a dusty factory floor).

Legal restrictions arise from any of the provisions, by which the activities of the organisation are subordinated, ranging from the legal form, the industry-specific regulations, to the detailed provisions (such as the Law on Electronic Signatures).

The assurance of cooperation from users (employees of the organisation) is a major factor in the success of the project. Any limitations in this area include both the level of competence of members of organisations and their technical expertise, as well as the technical and communication skills. Moreover, introducing IT to an organization is often associated with the fear of loss of employment or significance within the organisation, therefore the attitude of users towards changes can be a major risk factor for the project.

Sometimes there may be political constraints due to the type of organisation (public institutions) and political forces at the company itself (e.g. trade unions). It is preferable to quite early realise the potential sources of influence and potential conflicts.

The analysis of the limitations and conditions of the project leads to determining the project feasibility. It is good at this stage to draw attention to the relationships between the restrictions, if any, and to consider their potential impact on the course and the effects of the project.

The final stage of the feasibility study (the conclusions and recommendations) is reduced to the decision whether the project should be attempted or not. In the case of a positive decision, the project budget and duration should be defined. It is also good to consider the potential preparation of the organisation to the overall project.

The feasibility study is the shortest stage of design work, but taking the wrong decision at this stage will cause serious problems later in the design work, and can even cause complete failure of the project (especially the correct definition of the project's objectives).

The phase of the analysis of the needs is devoted to a detailed examination of the existing situation of the organisation. It is sometimes referred to as a requirements analysis. All work is devoted to identifying information problems and needs. The starting point is usually the analysis of the existing IS. IS analysts try to learn as much about the current functioning of the enterprise, using various data collection techniques, such as:

- Interviews (individual and group)
- Survey (anonymous, named)
- Observations (open and covert, and passive and active)
- Studying the documentation (gathered and used by the organisation)
- Experiments (simulations of situations that cannot be observed in the actual activity)

All observations have to be documented on an ongoing basis in the form of documentation methods and techniques presented in Section 2.3 (such as data flow diagrams, methods of describing the logic of processes, data dictionaries).

An important success factor of the analysis is ensuring the cooperation and involvement of system users. If they are convinced of the need for a change, they will also tend to make a fair presentation of their problems and requirements regarding the created IS. In the case of negative attitudes of staff to the solutions introduced (fear of job loss, increased requirements for qualifications, etc.), the whole project is likely to end in failure.

Logical design is a phase intended to create an organisational solution, a logical model of the new IS. Using the same methods of documenting as at the analysis stage, the IS analysts and designers try to develop solutions that best meet the information needs and solve the problems of the organisation. Emphasis is put on requirements relating to data, the method of their presentation and transmission. However, the technical process of collecting, processing and storing data is not decided upon. From the point of view of the logical design, it is essential that the user has access to customer data, for example the name, address, and fees. The way the data will be stored and in which sets is a part of the detailed (technical, physical) design. Similarly, the description of the logic of processes emphasises organisational procedures, decision rules, and not the algorithms from the perspective of software. The solution created by the analysts and designers must be subject to verification on an ongoing basis by qualified system users, in order to be sure that the IS created matches the actual requirements of the organisation. The logical model created after the acceptance of the management and key system users is then subjected to elaboration at the detailed design stage.

The detailed design, also known as a physical or technical design, is the actual processing of solutions at the organisational level into the requirements (capabilities and limitations) related to the IT. Organisational procedures are presented in the form of processing algorithms (flowcharts, pseudo code, etc.), data dictionaries and models are refined (databases are standardised, the structure of individual tables and the relationships between them are determined), application inputs and outputs are designed, as well as the user interface. This stage is in fact the actual designing of the software, but the previous analysis and preparation of the logical design allows for adapting of the software to the actual needs of the organisation.

The implementation stage, although associated with the implementation of the solution, is dedicated to creating software. The requirements contained in the detailed design are written in the selected programming languages, database management systems, or application generators. The code documentation should be created in parallel to the development of software code, allowing for later alterations in the software. At this stage, the technical documentation of systems maintenance is created for network and database administrators and hardware maintenance staff, as well as operating manuals for software users. Created applications are also tested, first at the level of syntactic correctness of the code, then the accuracy of generated results, resistance to erroneous data, interoperation with other modules and data sharing, through testing of subsystems, to testing the whole software. The test of the entire system is first carried out using artificial data, the final test using real data is performed in actual conditions of system operation and is called operational testing. After correcting any errors found, the system is put into operation and thus its use in the organisation begins.

The above implementation stage is characteristic for the creation of dedicated software (written to a specific customer order). In practice, organisations often use ready-made software, which is adapted to their needs. In this case, the implementation involves coding to some extent and parameterisation of existing applications in regard to the requirements of a particular organisation.

2.2.2. The system life cycle

The information system within an organisation plays the role of a "nervous system", allowing communication between various components of the organisation. The effectiveness of the system is not constant and is associated with the system life cycle (Fig. 2.16). The system does not reach its full effectiveness when implemented, because employees must first become familiar with new concepts and procedures, learn to use the implemented software. In addition, the implemented software has to gain access to data in order to be able to generate any results. Some data can be entered using historical data, but almost always new software includes new features for which data have not been collected yet. Thus, the effectiveness of an information system in the initial stage increases, reaching a point of maximum use of its (functional) capacity. It should be noted that use of the information system takes place for months or years, during which the market environment and the organisation itself are changing. If the IS is not modified, then at some point its potential to support the activities of the company is reduced, which in other words means a reduction of its effectiveness. To maintain this effectiveness at a sufficient levels, modifications are made to the system, the existing solutions are developed, new capabilities, including technical, are added (e.g. software features added, hardware is replaced, etc.). There may however come a moment when the development of the existing system will become irrational (too time-consuming or too expensive in relation to the benefits). Then the whole system should be replaced.

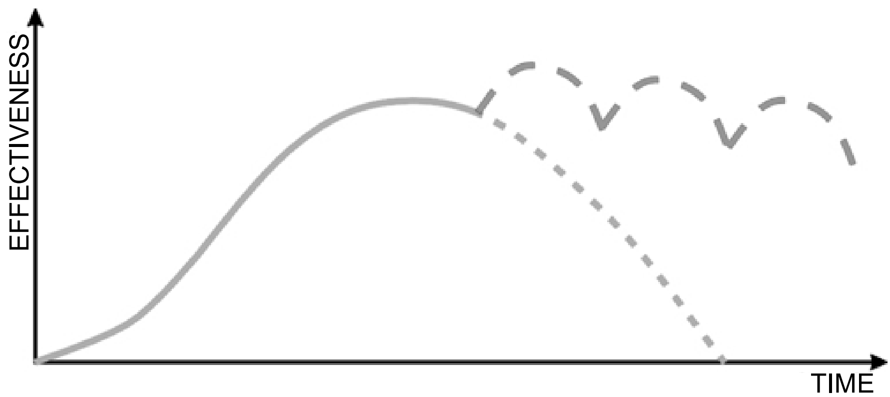

Fig. 2.16 The system life cycle
Source: Own work

The lifetime of the system depends on the variability of organization's environment, quality of the initial solution (strategic adaptation of the IS designed to the future needs of the organisation) as well as opportunities for the development of the IS (the so-called openness of the software). In particular, the variability of the environment is an important factor in shortening the life cycle of the system and forces the necessity to introduce new solutions. The more stable the environment of the organisation, the longer the lifespan of the existing IS.

2.3. DESCRIPTION OF SELECTED METHODOLOGIES, METHODS AND TOOLS

2.3.1. The evolution of information systems analysis and design methods

Analysis and design of information systems is an issue on which the scientific and business circles have focus their attention since the first applications of computers and microcomputers, that is, since the 50s of the twentieth century. During thvis period, information systems analysis methodologies evolved from very informal and intuitive to structured and formalised, supported by dedicated computer tools – CASE (Computer Aided Software Engineering) systems. First we need to define the notion of methodology in order to better understand the content conveyed in this section. According to the Dictionary of Polish Language PWN33, a methodology is:

A set of rules regarding the method of performing a given task.

To apply this to information systems analysis, it is assumed that the methodology of creating an information system is (Wrycza, 1999, p. 11):

A coherent, logically structured set of methods and procedures of a technical and structured nature allowing the executive team to implement the system life cycle.

Methods of a technical nature include models, tools and techniques used in the production cycle of the software. In the remainder of the chapter, most popular methods of analysis and design of the SI will be characterised according to the chronology of their popularisation:

- Narrative functional specifications
- Structural
- Prototyping
- Soft systems methodology
- Object-oriented
- Process-oriented

Currently, many different analysis and design methodologies are used worldwide. Corporations and research centres create their own so-called proprietary methodologies, but the vast majority of them are based on the above "traditional" approaches.

Narrative models were created during the 70s of the twentieth century. Using the natural non-formalised language, analysts have documented the requirements, which then became the basis for the development of systems. Narrative models have many flaws, which include (Yourdon, 1996):

- *Ambiguity* is typical for a natural language, and when analysed, it leads to a different understanding of concepts by the user, the analyst, the designer or the programmer.
- *Redundancy* – the occurrence of the same aspect in many places in the specifications.
- *Monolithic character and lack of structure* – it is difficult to select a single aspect of the system, and the whole document must be examined in order to ascertain the selection.
- *Labour-intensive maintenance* – the two previously mentioned flaws caused a very laborious and error-generating process of updating analytical

[33] http://sjp.pwn.pl/szukaj/metodyka

documents, which in turn led to their gradual obsolescence. The greater the range and the time of analysis, the less useful the specifications became.

Narrative methodologies led to the formation of a large volume, but not very relevant and hard-to-use documentation, which in the final effect was rejected and ended up unused.

Structural methodologies

With the increasing popularity of computer technology, the growing importance and complexity of information systems and the emergence of structural programming languages (i.e. *Pascal*, *C*), the creation of more formal and unambiguous methodologies was initiated – called structural methodologies.

The initial assumption of structural methodologies concerned (Yourdon, 1996):

- *Graphical modelling* – according to the saying "a picture is worth a thousand words", specification consisted of a set of diagrams. Diagrams were provided with text descriptions, but always very formal and concise.
- *Manifoldness* – a structural model showed the information system from different perspectives (convergent in the whole model), which could be easily analysed separately. The main structural perspectives, and also the main paradigm of this method, divided the information system into the processing perspective – i.e. the process model, and the data perspective – i.e., the data model. The tools in each model and the timetable for their implementation are described in the section "Structural IS analysis and design methodologies".
- *Minimum redundancy* – so that the modification of documentation resulted in changes to few very specific parts of the specifications only.

The main structural methodologies of this period include the STRADIS methodology (Gane and Sarson, 1979). The very popular SSADM methodology (Structured System Analysis and Design Method) was built based on the structural approach. This methodology was developed at the request of the British government and was used as standard for projects in the public sector (Beynon Davis, 1999).

Structural analysis has grown rapidly until the early 90s of the twentieth century. Modification of structural methodologies at the end of the eighties can be called a modern structural analysis, after the book by E. Youtdon titled *Modern Structured Analysis* (Yourdon, 1996). This publication is in the original version was released in 1988. The author combines the structural advantages of previous methodologies and replaces flaws detected during several years of practice with his own solutions. The result is a new methodology, called the modern methodology, or simply Youdon's methodology (YSM). Modern structural analysis brought the following new elements:

- Moving away from the construction of the existing physical model of the system (as time-consuming and preserving old solutions) and emphasis on logical and physical modelling of the target system.
- Clear separation of logical and physical models by introducing the concepts of the basic (logical) model and the implementation (physical) model. The first one was created in the stage of the structural analysis and answered the question of how the system should operate in order to meet user requirements. The second one was created in the design stage and showed how to implement the system in order to meet the requirements described in the basic model.
- The perspective of system behaviour over time was taken into account, very helpful in modelling real-time systems. A diagram of the system's state was created and extensions in the process model were introduced.
- Greater emphasis was placed on data modelling. The data structure diagram was replaced with a much more powerful (and popular to this day) entity relationship diagrams (ERD).
- The approaches to the direction of system analysis were changed. Classical structural analysis preferred the top-down approach, and so constructed diagrams at the highest level (most general) and then decomposed them by building subsequent more detailed levels. The top-down approach led to a number of inconsistencies and omissions of important lower level processing. Yourdon's approach consisted of specifying a list of events to which the system must respond. Then, processes handling individual events are designed. Modelled basic processes are incorporated into the processes at higher levels. This approach is called a bottom-up approach. More details are included in Section 2.3.2.

In the structural methodologies and subsequent ones, which use graphical modelling, a milestone was the implementation of programs such as CASE, supporting graphic techniques and having built-in repositories and certain knowledge on analytical methodologies. The problem facing the makers of these tools was the lack of standardisation and agreement between the methodologies. This situation was undoubtedly improved by the UML language, which is described in more detail in the next chapter, dealing with object-oriented analysis methods.

Prototyping

Prototyping is an approach to the implementation of the IS development cycle, it can also be used as a supporting analytical technique. Prototypes of the system functionality are built in order to achieve better communication between user and supplier. Prototyping can be used within the analytical methodologies e.g. in the structural methodology.

Criticism of the structural approach and the transition to object-oriented methodologies

The basic element of criticism of the structural approach was the negation of the assumption about the separate discussion of the aspect of processing and data. P. Coad and E. Yourdon (1994) in their book, *Object-Oriented Analysis* point out that in large projects there is a trend to separate the thematic areas and analysis teams into two areas: data model and process model (meaning the processes). The methodology itself did not provide appropriate tools for synchronising the two models that have developed on their own, and over time began to be inconsistent. Lack of synchronisation of the models led to creation of a comprehensive documentation which was inconsistent and difficult to use. As a result, problems were transferred to the subsequent stages of the cycle. Another gripe of structural methodologies was the lack of continuation of analytical tools used in subsequent phases i.e., in the design and implementation (as opposed to object-oriented methodologies, whose description follows).

Object-oriented methodologies

The end of the 80s of the twentieth century was dominated by structural methodology. Its development has led to the creation of many derivative methodologies. These methods, based on the assumption of separation of the data and processing models, have failed to eliminate the problems discussed in the previous section. At that time, in the area of implementation, structured programming was being replaced by object-oriented programming languages, such as *Simula*, *Smalltalk* and finally *C++*. Object-oriented languages above all provided a greater stability, and a safer and easier ability to modify systems created. Disadvantages of the structural approach and advantages of the object-oriented programming led to the transfer of object-oriented paradigms to the analysis and design phases. The use of encapsulation (one of the main premises of object-orientation) provided a combination of analysis and data processing areas in one place in the description – the object. This solution eliminated the fundamental flaw in the structural approach. Object-oriented methodologies after the initial phase of the so-called "methodology wars" became very much unified and provided with common analytical and design techniques. A manifestation of this unification was the creation of the UML language (Unified Modeling Language) and the OMG organisation (Object Management Group), which worked on its development. The object-oriented approach is the basis for the creation of comprehensive software development methodologies such as the RUP (Rational Unified Process) methodology described in the next section.

Soft systems approach

As a part of the soft systems approach, which takes into account the human, communication, cultural, and organisational aspects of the development of IS, methodologies were also defined, the most famous of which is the soft systems methodology. Soft systems approach is described in Section 2.1.

Process approach

A relatively new approach to the analysis of IS is the use of the concepts of management through business processes as a paradigm, and the implementation of the IS as its consequence. This approach is discussed in Section 3.1.

2.3.2. Structural analysis and design methodologies

2.3.2.1. Structured system development cycle

The evolution of structural methodology has been described in previous sections. It was pointed out that one of the most recent publications related to the structural analysis is *Modern Structured Analysis* by E. Yourdon (Polish edition *Współczesna analiza strukturalna*) (Yourdon, 1996). In this book the author points out the flaws of the classical structural analysis and proposes his own solution to these problems by proposing a comprehensive methodology called the Yourdon Structured Method (YSM). The methodology has been selected as a methodical basis in this section of the chapter. The description will begin with the structured system development cycle.

The structured development cycle of the IS (Fig. 2.17) is derived from the traditional waterfall approach, however, it improves two of its flaws:

1. The bottom-up method of system implementation,
2. Departure from the strict sequential transition between successive stages.

E. Yourdon suggests using a top-down sequence of coding, where the superior modules are coded and tested first, then the detailed modules are created and tested. Another very important assumption of the structured cycle is the possibility of a parallel (not sequential) implementation of individual phases. A parallel start of the phase of preparation of acceptance scenarios, system documentation, and design stages after the structural analysis is also suggested. One of the premises is also the iteration between implementation and testing or analysis.

In the Yourdon approach, the analysis and design stages are precisely separated. In the first stage, the so-called basic model of the system is created, which defines what the system should do to meet the requirements. The implementation model is formed in the design stage, showing the technical aspects and determining how the system should be created.

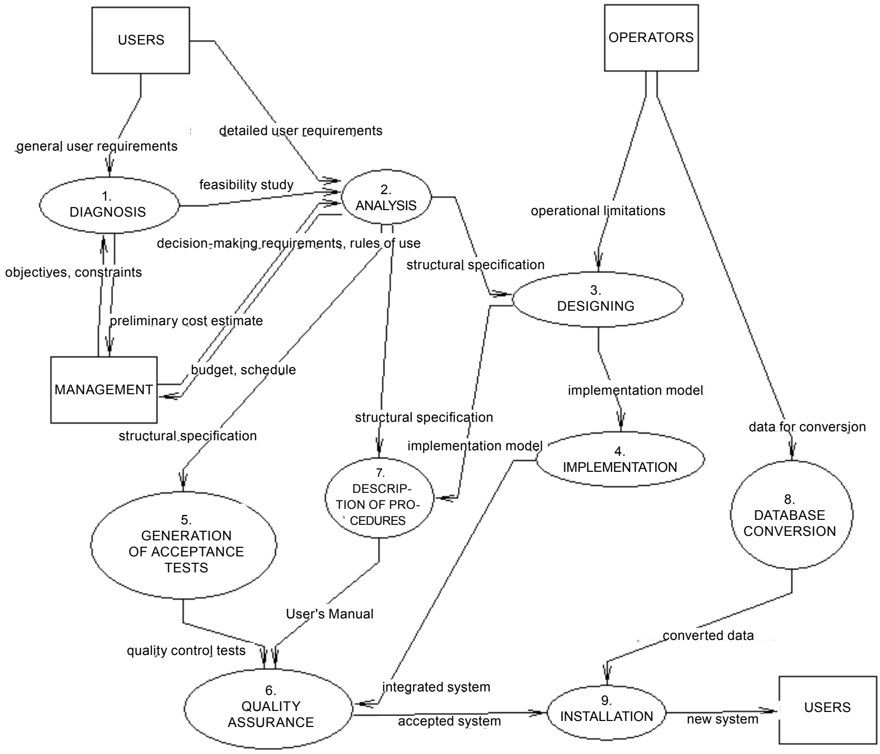

Fig. 2.17 Structured IS development cycle
Source: Own work based on (Yourdon, 1996)

2.3.2.2. Three aspects of the system

All structural methodologies, including the Yourdon methodology, are characterised by the manifoldness of the approach to the description of the IS. The most important structural aspects are data and processing, i.e. the (information) processes. Additionally, Yourdon discusses the aspect of system behaviour in time, which is dedicated to real-time systems, in his methodology.

Every aspect has the right graphic and textual tools, which will be described below, in the discussion on the steps and tools for structural analysis.

The basic model, which is the result of the structural analysis stage, is divided into two sub-models:

1. *Environmental model* – showing the elements of the environment which the modelled system will exchange data with. This model defines the

border of the system being created. The environmental model includes the following tools:

- A description of the system objectives.
- A context diagram – a data flow diagram (the technique will be described in the next section), whose characteristic feature is that it has only one process, which maps the analysed system. The role of the context diagram is to show what elements are in the system's environment.
- An event list – showing the real-life situations of the analysed organisation to which the system will have to react, e.g. a customer wants to rent a film.

2. *Behaviour model* – showing the required internal behaviour of the system necessary for proper interaction with the environment. The behaviour model uses the following tools, taking into account the basic aspects of analysing the system:
   - In terms of processes (process model):
     – Data flow diagrams (DFD)
     – Data dictionaries (DD)
     – Process specifications (PSPEC)
   - In terms of data (data model):
     – Entity relationship diagrams (ERD)
   - In other terms – time characteristics of the system:
     – State transition diagrams (STD)

The following sections describe the structural analysis tools using which the behaviour model is created (an exception is the aspect of system's behaviour over time, which will not be discussed in this textbook).

2.3.2.3. Data flow diagrams

A data flow diagram shows the processes that are implemented by the modelled system. DFD is a modelling tool that allows to illustrate the system as a network of functional processes, connected by "data streams" (i.e., flows) and "containers" (i.e., stores) of data (Yourdon, 1996). The basic elements of the diagram are processes, data flows, terminators and data stores.

Notation

The Process (Fig. 2.18) – converts input data to output data. It is the only DFD element that can perform operations on data. Since processes perform a given work, their names should contain an imperative verb and an object, indicating what will be the subject of this work, e.g. accept orders, order goods.

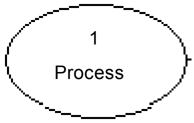

Fig. 2.18 Notation of the process
Source: Own work

The use of the circle (the proverbial bubble) as a symbol of the process is not a generally accepted notation. In other notations, the process may be rectangular with rounded corners.

The Terminator (Fig. 2.19) represents persons, groups of people, or other systems that will communicate with the system being modelled. Of course, communication must be in the scope/responsibilities of our system. A terminator is always named using a singular noun.

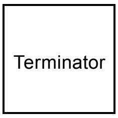

Fig. 2.19 Notation of the terminator
Source: Own work

The store (warehouse) (Fig. 2.20) shows the data, which the information system stores for a period of time. Stores are the “memory” of the system, no other element can store data. The names of the data stores should be plural nouns.

Fig. 2.20 Notation of the data store
Source: Own work

The Flow (Fig. 2.21) represents data passed between the remaining elements of the diagram (system). Flows are data in motion. Flows are marked with arrows with the explicit indication from which the element of the DFD to which the data packet is sent.

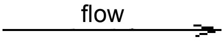

Fig. 2.21 Notation of the data flow
Source: Own work

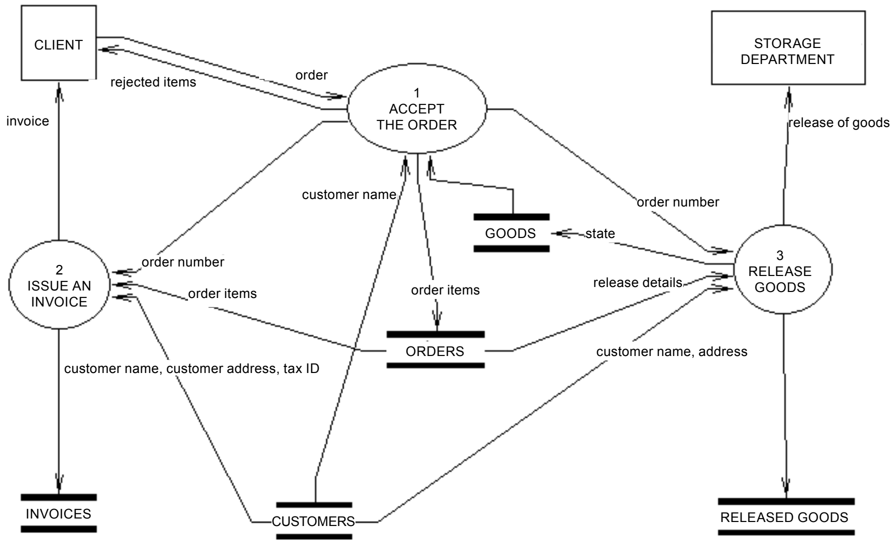

Fig. 2.22 Diagram of a trading system
Source: Own work

Figure Fig. 2.22 shows the process of handling orders in a trading company. It was assumed that the customers are registered by another process (outside the scope of the diagram) before placing an order. The second assumption is that the storage department is beyond the scope of our system.

Multi-level diagrams

Complexity of a data flow diagram is the maximum number of processes described in one diagram. A rule is assumed that six to seven processes in one diagram is the maximum amount. Above this value, the reader will have difficulty understanding the message. Seven is the number of processes sufficient to model a small system. In large systems, there are dozens, or even hundreds of processes. The solution to this problem is multi-level diagrams, in which we can decompose a large system to the main processes, then create their developed views at lower levels. The example of decomposition and a multi-level diagram is shown in the analysis of the film rental system, presented at the end of this section.

2.3.2.4. Data dictionaries

A data dictionary is a text tool describing all data elements of the modelled system. These elements are derived from data stores and flows in the DFD and the relations in the entity relationship diagrams ERD. A data dictionary is a very important element of communication between users, analysts, and especially designers and programmers. The latter may not come into contact with the users, while they must clearly understand the organisation's specific language.

The data dictionary contains two types of elements:

1. Elementary data – are not further divided, e.g. age, city, height
2. Data packets – sets composed of elementary data, such as data warehouse with data on customers, readers, etc.

In a dictionary, elementary data are defined differently from the data packets. For elementary data, we describe the context (i.e. the importance for the organisation) and the units and limit values, e.g.

```
Tonnage = * the weight of the vehicle entering the facility *
              * unit: tonne, range: 1-25 *
```

Data packets are defined by the context and the description of elementary data the packet is composed of. The following is a definition of a data packet describing a reader who uses the services of a modelled library system:

```
Reader = * a person who uses our library *
* = card number + surname + first name + social security number *
```

The convention of this notation comes from the Yourdon methodology. Most CASE tools have a so-called model repositories, where data elements used in diagrams appear as it were automatically. A very important feature of the data dictionary is an alphabetical arrangement of the elements appearing there.

An example of a data dictionary built in the form of a spreadsheet table is shown in the example at the end of this section.

2.3.2.5. Process specifications

The data flow diagram shows the processes, their inputs and outputs, but no information about the processing algorithm within the process. Other textual, tabular, and diagram tools for process specifications are used for this purpose. The most popular structured methods of process specification are:

- Decision tables
- Decision trees
- Initial and final conditions
- Structured English language
- Structural English language

This manual will describe the last of these methods – the structural Polish language, otherwise known as pseudocode. Descriptions of other methods can be found in the book by E. Yourdon (1996).

On one hand, pseudocode is structural, meaning sentences should be built with it using syntax which is similar to the representation of the program's statements, i.e. a statement + parameters of its call. On the other hand, it is the English language, because all the instructions and parameters in our specifications will be in English.

The four main principles of pseudocode are:

1. Pseudocode statements are verbs in English. It is the best to use the list:
   - Operations on data from the flows and stores: LOAD (SELECT, DOWNLOAD), DELETE, SET, CREATE, SAVE, ADD, SORT
   - Arithmetic operations: CALCULATE, ADD, SUBTRACT, MULTIPLY, DIVIDE
   - Communication operations: DISPLAY, SEND
2. Only data elements associated with our model, that is located in the data dictionary, can be used as statement parameters.
3. Conditional functions can be used in pseudocode. They describe the alternative action if the required conditions are met (IF-THEN, CASE, DO-WHILE). ###Conditional statements are in English and it is the only exception to the rule given earlier.
4. Local terms are data objects defined and used in the specification of a single process, as indirect calculations or loop counters.

Example of process specification using pseudocode is shown in the example at the end of this section.

2.3.2.6. Data model

One of two approaches can be used in order to create a data model:

1. In the first one, the top-down approach, we analyse the "scenario of the life of the organisation" in search of objects which the organisation wants to "know" about and the relationships between them. It is a conceptual approach (Baker, 1996).
2. In the second one, the bottom-up approach, we analyse specific data sets. The starting point is real data sets, whose sources can include:
   - Operating data sets – electronic or paper (forms or records)
   - Information system documentation
   - Data administrators

In this chapter, we will show the first approach. The second approach is usually an element of the lecture in the "Databases" course where we talk about the standardisation process.

Entity relationship diagrams (ERD) are used to present the data structure.

An entity relationship diagram is a network model that supports the arrangement of data stored in the system on a high level of abstraction (Yourdon, 1996). The author of this diagram is P. Chen. Entity relationship diagrams are so popular that as many as seven naming notations have been developed. This chapter uses the Chen's notation and the notation implemented in the *CASE Studio* tool.

Notation

The ERD diagram has three elements:

1. Object
2. Attribute
3. Relationship

Objects (or entities). Objects are the elements which interest the organisation and which exist in the real world of the organisation.

An object is an important thing or object, real or imaginary, information about which should be known or stored (Baker, 1996).

In the conceptual approach in the scenario of the life of an organisation, potential objects are nouns which occur in them. The main categories of objects are:

- Physical objects such as: employee, department, patient, flight, seat, aircraft, machine, schedule
- Events/concepts such as: surgery, car repair, transaction, order

Attributes. Attributes are the information we need which are related the objects (Baker, 1996). The attributes are the details related to identifying, classifying, determining the quantity, expressing the state, or any descriptions of the object. An object is an aggregate of attributes.

Relationships (or relation). A relationship is a named, important link existing between two objects (Baker, 1996).

In the conceptual approach, we are looking for sentences in the scenario of the life of an organisation, where there are two objects interacting with each other and it is possible to name this relationship. An important criterion for the recognition of this cooperation as a relation is the importance to the scope of responsibilities of our system. Sample sentences:

```
Client orders GOODS
READER lends BOOK
EMPLOYEE performs a TASK
```

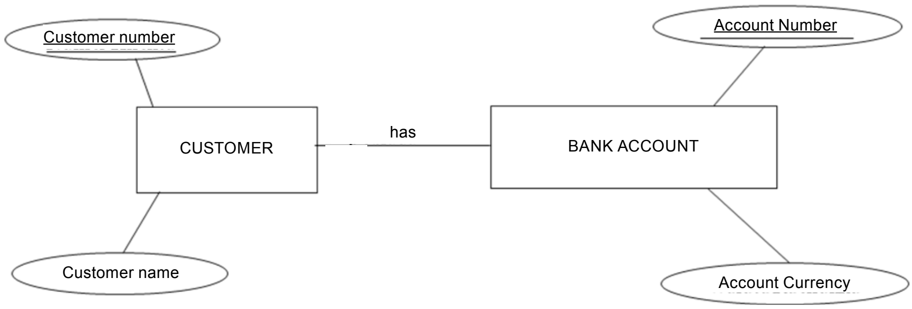

Fig. 2.23 Entity relationship diagram (ERD). Chen's notation
Source: Own work

ERD notation according to Chen (Fig. 2.23). Unselected objects are rectangles. Attributes are entered in ellipses. The relation is mapped with the line joining the objects. The relation is named.

The following describes the key attributes of a relation:

- Participation (optionality):
  - Required if all instances of an entity must participate in the relationship – the first relation in Fig. 2.24
  - Optional, if there is at least one instance of an entity that does not participate in a relationship – Fig. 2.24

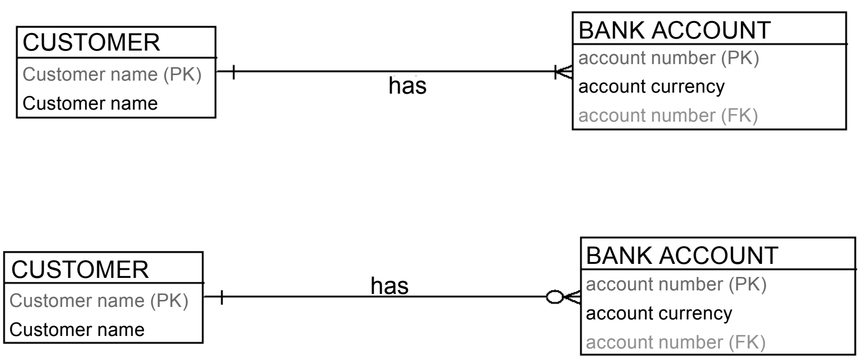

Fig. 2.24 Entity relationship diagram (ERD).
Notation of the required attribute
Source: Own work

- Cardinality (degree) – refers to the number of instances of objects on either side of the relation. The types of cardinalities on both sides of the relation: 0,1,N (where N means many).

Expanded ERD diagram is shown using the example of the film rental system in the next section.

2.3.2.7. The design of an information system of a sample organisations using structured methodology

Description of the situation of a company.

Company name: *Cool Film* Rental Shop

The business owner plans to open a film rental shop in a large shopping centre. In the beginning, he will only be renting films on DVDs and Blue-Ray discs. In the future, he plans to open a small place next to the rental shop, where the customers will be able to watch the films rented. In particular, he wants to create a place where parents could leave their children where they could see a video when their parents are shopping.

CFRSIS objective

The objective of the *Cool Film* rental shop information system (CFRSIS) is to collect information about the rental shop's collection and operations related to renting. The introduction of the system should lead to:

- Handling the rental and the return of a film by an already registered customer in less than 1 minute.
- The ability to control the demand reported by the customers for a new film appearing in the catalogues of distributors.
- At any time, it can be found where the disc available in the rental shop's offer is located.

CFRSIS context diagram

The context diagram presented at Fig. 2.25 is to determine what elements from the environment of the system analysed will interact with this system. In the analysed example, the three main elements of the system's environment are indicated:

1. Customer
2. Distributor
3. Owner

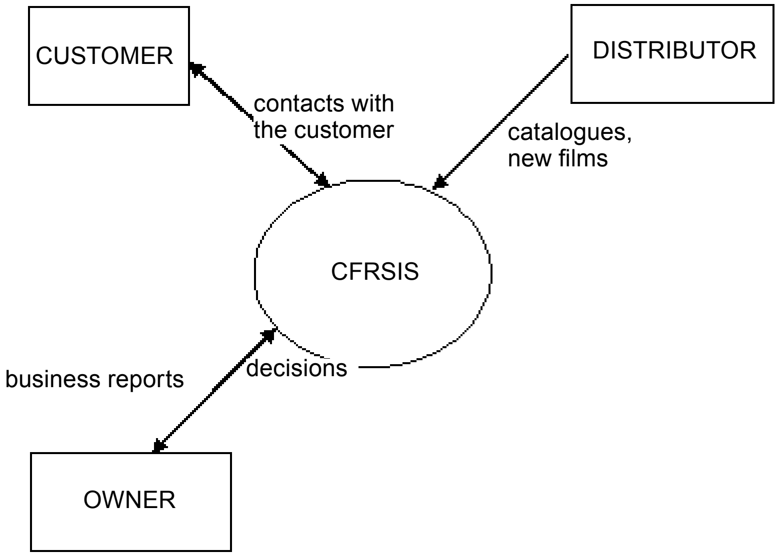

Fig. 2.25 The context diagram of the Cool Film rental shop information system
Source: Own work

CFRSIS event list

When analysing the current operations of the film rental shop, events are recorded that will fall within the scope of operation of the proposed IT tool.

- Customer wants to register.
- Customer request information about a film.
- Customer rents a film.
- Customer returns a film.
- Customer requests information on the current account balance.
- Customer reports information about the inability to return a film.
- Distributor sends a catalogue.
- Distributor sends a new film.
- Owner needs a (weekly) report on current arrears (T).
- Owner needs a (monthly) report the number of films rented and the number of new customers (T).
- Owner needs a (monthly) report on the number of requests for new films (T).

Behaviour model

The following is the behaviour model of a sample organisation – the Cool Film rental shop. The behaviour model starts a full (in this case) two-level data flow diagram. The top level – called the "zero" level – represents the main information processes of the organisation. The zero level diagram for a sample organisation is shown in Fig. 2.26.

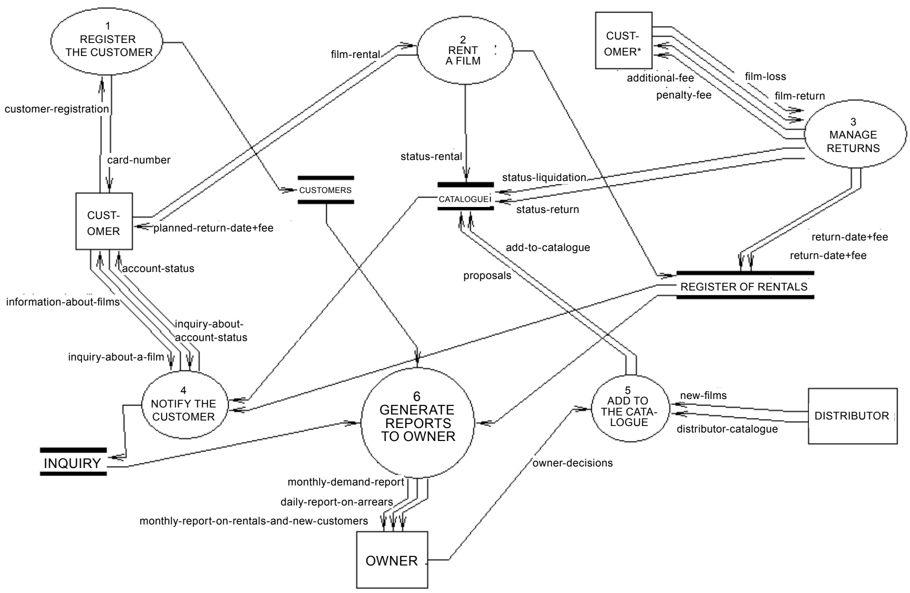

Fig. 2.26 Level 0 DFD for the Cool Film rental shop information system
Source: Own work

Next, the main processes are decomposed – shown in more detail in lower level diagrams (the first level in our case). First level diagrams are presented in Fig. 2.27, Fig. 2.28, Fig. 2.29 and Fig. 2.30.

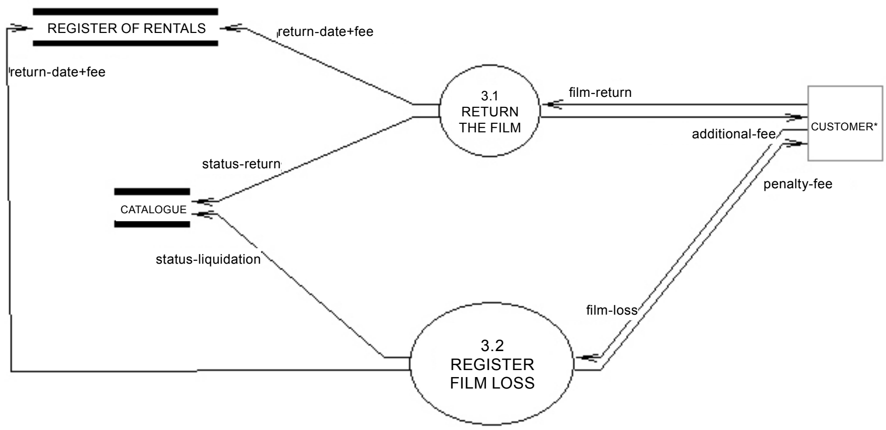

Fig. 2.27 Level 1 DFD for process No. 3 Returns Management

Source: Own work

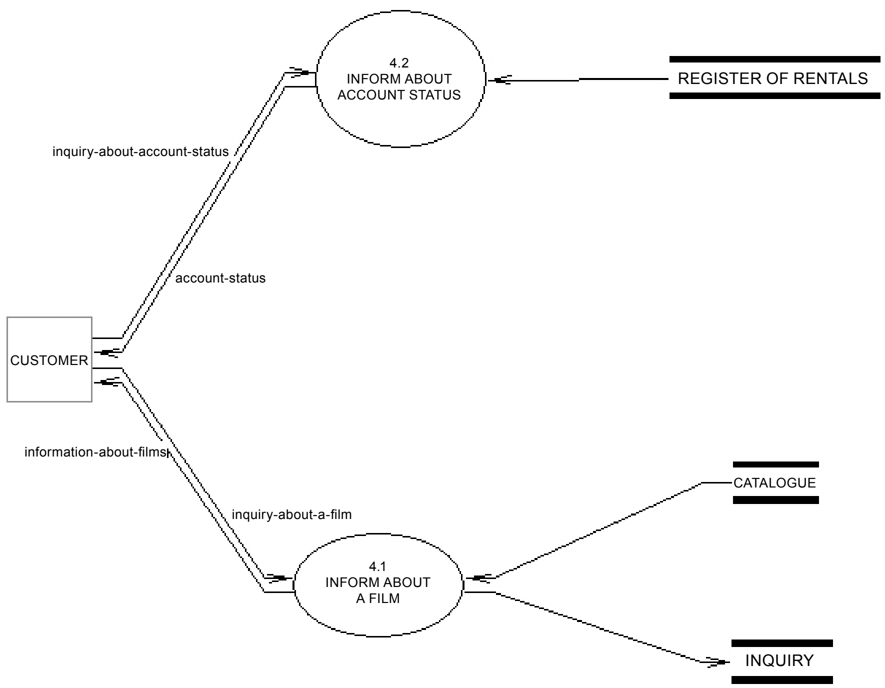

Fig. 2.28 Level 1 DFD for process No. 4 Customer Notification

Source: Own work

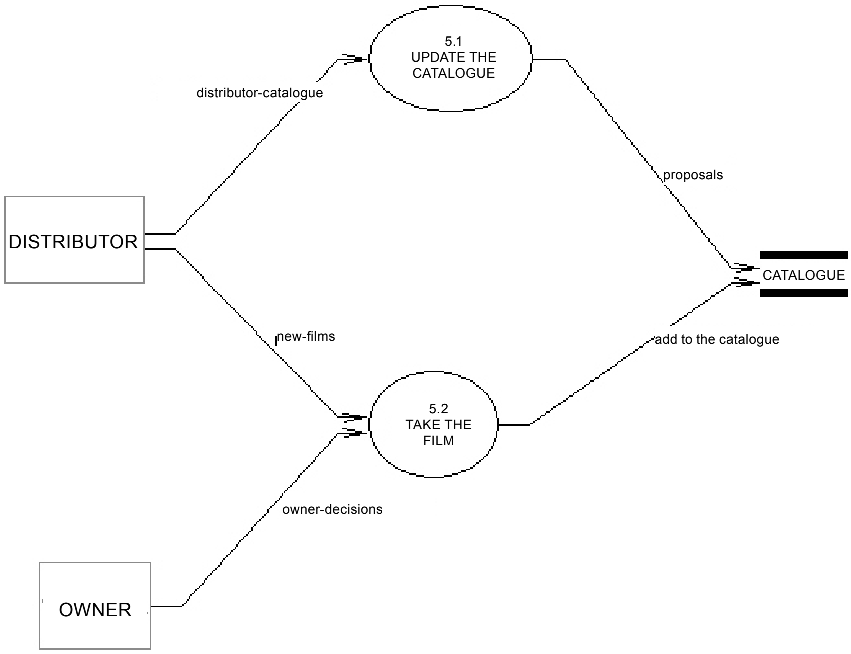

Fig. 2.29 Level 1 DFD for process No. 5 Returns Management

Source: Own work

Example process specification:

Process 2 – RENT A FILM (specification using pseudocode) is as follows:

```
BEGIN
 SET signature=signature with film-rental
 SET card-number=card-number with film-rental
 SET rental-date=system-date
 SET planned-return-date=system-date + 1
 SET return-date=empty
 LOAD film from CATALOGUE with signature=signature with film-rental
 SET fee=daily-rate with CATALOGUE
 CREATE record rental=signature+card-number+rental-date+
 return-data+planned-return-date+fee
 ADD record rental to RENTAL REGISTER
 --update data in CATALOGUE
 SET status='W'
 SET status-date=planned-return-date
 SAVE film to CATALOGUE
 SEND planned-return-date+fee to CUSTOMER
END
```

Data dictionary (part)

The data dictionary is a supplementation of the analysis processing aspect, which "translates" and specifies all the components in the diagrams and specifications of the behaviour model. Table 2.6 below shows a part of a data dictionary for the organisation analysed in the example – Cool Film rental shop.

Table 2.6 Part of the data dictionary of the CFRSIS system

Elements of the dictionary	Context	Definition	Limit values	Unit/format
Actors	*list of the main actors in the film *			Text
purchase-price	*cost of film purchase*		0-10000	PLN
monthly-report-on-rentals-	*a report showing: how many rental operations took place in the given period - overall with the details of the 30 most and least popular titles and additional information - how many new customers signed up*	"The amount of rentals in the period:"+ rentals-sum+ {title, rentals-sum}+"The number of new customers:"+sum-new-clients		
monthly-demand-report	*the report shows interest (expressed in number of requests) in new films in the catalogue for a period of 1 month*	{title+directors-name+distributor+ number of requests}		
weekly-report-on-arrears	*a list of customers with films, which they should have returned by the time of the report (and have not)*	{last name+ first name+ (phone)+ (email)+ address+ signature+ rental-date+ planned-return-date}		
liquidation-date	*date of liquidation\loss of the film by the customer*			date (yyyy-mm-dd))
release-date	*planned release date - the launch of the film*			date (yyyy-mm-dd)
registration-date	*data specifying when a customer registered in our rental shop*			date (yyyy-mm-dd)
date-of-birth	*customer's date of birth*			date (yyyy-mm-dd)
rental-date	*date of entering the rental transaction*			date (yyyy-mm-dd)

Source: Own work

ERD diagram (CASE notation)

Fig. 2.30 illustrates the ERD diagram showing the data structure of a sample organisation. It is clear from the diagram that the "customer rents the film" relation is an essential element of the data structure of the analysed system.

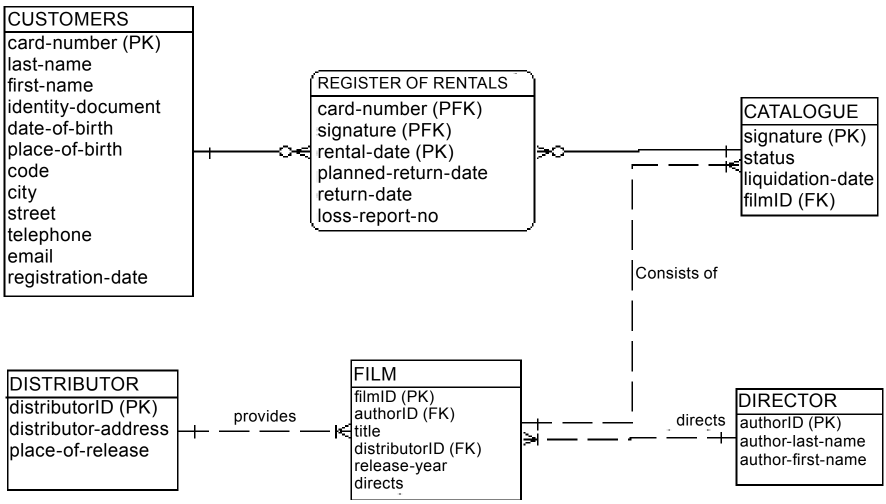

Fig. 2.30 ERD diagram for a film rental shop system

Source: Own work

The creation of the model of the system's behaviour was completed in the data model. Remember that in the structural analysis, there is also the aspect of the system changes over time, which due to the nature of the example system will not be created. Those interested in this aspect can find detailed information about the subject in the E. Yourdon's book (1996).

2.3.3. Object-oriented analysis and design methodologies

2.3.3.1. Object-oriented paradigm

Criticism of the structural approach and the popularity of object-oriented programming languages initiated the adaptation of the main premises of object-orientation in the analysis and design of IS. The most general principles of object-oriented analysis were set forth by P. Coad and E. Yourdon in their book *Object-Oriented Analysis* (Coad and Yourdon, 1994), originally published in 1991. The authors mention three methods of organisation, which people use to get to know the surrounding complex reality:

1. The differentiation of cognition into individual objects and their attributes – such as the distinction between a tree, its size or a man, his or her name, and age.
2. The distinction between whole objects and their component parts – such as a car and its engine, bodywork and wheels.
3. Creating classes of objects and distinguish between them – for example, the people class and the cars class.

These general principles for understanding the complexity and the development of object-oriented programming provided the basis for the identification of several basic paradigms of object-orientation (Wrycza et al., 2005, p. 17):

- ***Object*** *– each entity – the concept or thing – having a meaning in the context of the problem solution in a particular field.*
- ***Class*** *– a generalisation of a set of objects that share the same attributes, operations, relationships and meaning.*
- ***Encapsulation*** *– diversifying access to the object disclosing to the surroundings only the information about its attributes and operations that are necessary for effective reference to the object in the system by means of messages.*
- ***Message*** *– a specification of the exchange of information between objects, containing orders for the execution of a specific operation.*
- ***Inheritance*** *– assigning of attributes and operations classes of objects based on the hierarchical dependence between them, allowing to gradually build more and more specialised varieties of classes.*
- ***Polymorphism*** *– the ability to give the same name to various operations and perform various procedures and actions through the operations with the same name. This feature helps to simplify the exchange of messages between objects of different classes.*

A very important conclusion drawn from the analysis of the object-oriented premises shown is to enclose attributes, relationships, and most of all operations of the object within its concept. Such an approach is the negation of the structural approach, which resulted in a clear separation between data and processes.

Based on the paradigms shown, an equation can be derived, which makes it possible to recognise whether the method, tool, or programming language used is object-oriented (Coad and Yourdon, 1994):

Object-oriented = classes and objects + inheritance + communication using messages

Based on the assumptions shown in this section one of the most important unification in the area of IS analysis and design was created, namely the unified modelling language UML.

2.3.3.2. The origin and evolution of UML

Late eighties and early nineties was a period of the so-called "methodology wars". Two approaches collided: the structural approach and the object-oriented approach. The structural approach was heavily criticised and supporters of object-orientation competed in developing their own notations and proclaimed their own methodologies. According to (Wrycza et al., 2005) the number of identified solutions has grown in these years from few to over fifty. In this methodology "turmoil", the producers of CASE tools had a very difficult time deciding what methodology they should integrate into their products. Among the many object-oriented methodologies, three of them had a huge impact on the crystallisation and unification of object-orientation. The authors have joined forces to start work, which will be completed by creating the unified modelling language UML.

These authors are:

- J. Rumbaugh, author of the OMT (Object Modelling Technique) methodology
- G. Booch, author of the OOAD (Object Oriented Analysis and Design) methodology
- I. Jacobson, author of the OOSE (Object Oriented Software Engineering) methodology

Unification of these methods began in 1994 when J. Rumbaugh joined Rational Software, the company where G. Booch worked at the time. Both authors started work on merging their own experiences into a unified methodology. Rumbaugh and Booch presented their unified methodology, which they called the Unified Method (version 0.8), at the OOPSLA Conference34 in 1995. Also, at this conference, it was announced that *Rational Software* acquired *Objectory*, where I. Jacobson was employed. From this point Rumbaugh, Booch, and Jacobson began joint work on the inclusion of the concept of use cases into the unified methodology. Also from this point, referring to the three authors as the "three musketeers" became common. After including the use cases in the unified methodology, the latter turned into UML, the unified modelling language. However, this was not the end of "methodology wars", because the creators of the other methods were not convinced of the superiority of UML over their own work. In this situation, an important role was played by the organisation called the Object Management Group (OMG), which affiliates the authors and users of object-oriented methods. OMG's task is to promote object-oriented methods in software development. OMG is an organization supported by the world's largest ICT companies. Two task forces

[34] International Conference on Object Oriented Programming, Systems, Languages and Applications.

were created under OMG. The first one was the Analysis and Design Task Force (ADTF), which worked on developing a common object-orientation standard, the second one was the Revision Task Force (RTF), which dealt with changes in the previously agreed standard. In January 1997, the UML version 1.0 was created at Rational Software. This version has been sent to OMG for review. Version 1.1 was created in July 1997. ADTF adopted version 1.1 as an OMG standard in November 1997. Subsequent versions have already been developed by the RTF work group. In 1998, version 1.2 is released. Subsequent versions of the standard, up to version 2.0, were published approximately once a year. However, the most significant changes are introduced in version 2.0. In this version, the set of available diagrams was expanded to 13, updating some of the earlier notations. Also, a universal format of the representation of XMI models was developed, based on XML. Version 2.2 was currently published. In this textbook, we describe the most basic and most important diagrams only, which exist in UML from its beginning.

At the end of the description of the origins of UML, its general definition is given (Wrycza et al., 2005, p. 20):

> *UML (Unified Modelling Language) is a graphical language for visualisation, specification, creating and documenting IT systems.*

This language is currently being developed by the Object Management Group. It is interesting to note that UML is not a methodology, but a language that can be used regardless of the methodology used for software development. Since UML grew out of object-orientation, it is best suited for object-oriented methodologies, however, some tools can also be used in other approaches.

2.3.3.3. The types of UML diagrams

As mentioned above, UML is a collection of graphical models, using which we can describe all aspects of the information system. UML version 2.0, on which this description is based, has thirteen diagrams, divided into four types. The following is a generic structure with the diagrams included in their structures:

- Structures
  - Class diagram
  - Object diagram
  - Package diagram
  - Diagram of combined structures
- Dynamics
  - Use case diagram

  - Activity diagram
  - State machine diagram
- Interaction
  - Sequence diagram
  - Communication diagram
  - Scheduling diagram
  - Interaction control diagram
- Implementation
  - Component diagram
  - Deployment diagram

The class, use case, sequence and state machine diagrams – as the most frequently used, are more accurately described in the following sections of this textbook.

Class diagrams

The concepts of a class and object are the key elements of object-orientation, therefore the class diagram is the central technique of UML and methodologies based on object-orientation. The class diagram belongs to the type of structure diagrams and describes the types of objects in the system and various kinds of static relations between them (Fowler and Scott, 2002).

Examining the concept of a class and an object, another version of the definition (the first one was presented in section 2.3.3.1) is quoted, this time taken from *Object-Oriented Analysis* (Coad and Yourdon, 1994, p. 50). Both definitions emphasize the domain and scope of the system.

An object is:

> *An abstraction of something in the problem domain, reflecting the system's ability to store information about this, the interaction with this, or both. An object is a capsule with the values of attributes, and services (operations) active on them only.*

A class is:

> *A description of the object or objects with a uniform set of attributes and services, containing a description of how to create new objects in the class.*

Building a model of system classes take place in an incremental manner and can be based on the so-called five-layer analysis model given by Coad and Yourdon (1994). Five successive layers are:

1. Subject layer – areas of the domain are divided into smaller fragments

2. Class/object layer – searched from the description of the problem domain
3. Structure layer – contains the relationships between classes
4. Attribute layer
5. Service layer

The individual layers will be described in subsequent sections. At the same time, the description of the messages is added, as an important element associated with object-orientation.

Subjects

The subject layer is particularly important in large complex systems, where the domain consists of many areas. The division into subject areas (subjects) is the solution to the complexity problem. In terms of the graphical notation, UML introduced a special type of diagram, the so-called package diagram, where the package can be a part of the whole system, i.e. one of the subjects.

Classes and objects

How to find object classes? Searching in the scenario of the organisation of classes and objects is one of the most important issues in object-oriented analysis. Coad and Yourdon (1994) mention a list of elements that should be analysed in search of classes and objects, and they are: things or events, equipment, roles played, operating procedures, locations of organisation units, relations, and other systems. The events stored, which are immaterial and are used for documentation of the history and dynamics of the problem domain, are a very important element of this list.

Notation

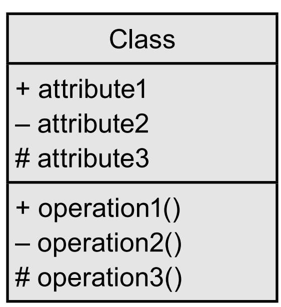

Fig. 2.31 Class symbol
Source: Own work

The class name is a singular noun or a noun and an adjective. The terms should be taken directly from the scenario, but abbreviations and acronyms should be avoided. The class symbol is denoted with a rectangle with three sections in the diagram. In order to improve clarity, the second and third section are optional in diagrams.

The first section has the class name, the second one has the list of attributes and the third the list of operations available in this class (Fig. 2.31).

Attributes

Attributes are certain data (system state), for which every object in the given class has its value (Coad and Yourdon, 1994).

How to look for attributes? In relation to the objects in the domain, we should ask the questions:

- What do we want to know about the object?
- How is the object described in the problem domain?
- In what states can this object be?

Notation

For the names of attributes, nouns or nouns with adjectives are usually used (prefixes, suffixes, abbreviations, and acronyms should not be used). An important element, to which Coad and Yourdon (1994) draw attention, is to treat the attribute as an atomic concept or a strongly related group of values. An atomic concept is a single value (indivisible) and the group of values is a set of naturally associated components, e.g. the address may consist of street, house number, town. The use of sets of values affects the readability of the diagram, especially at the conceptual stage.

The following is a list of key features that the attributes of objects have:

- Traceability – some attributes, like in data modelling, play the role of unique identifiers. In class diagrams, especially at the concept stage, we can assume that each object has an automatically generated attribute which is a unique identifier.
- Visibility – is a trait of attributes resulting from encapsulation. Encapsulation means the existence of attributes or services from which only the objects and services of the given class can benefit. These attributes are inaccessible (invisible) for other model classes. The three most important visibility symbols are:
  - Public (+) - means the availability of the attribute for the objects of all classes.
  - Private (-) - means the availability of the attribute only for the objects of the class to which the attribute belongs.
  - Protected (#) - means the availability of the attribute only for the objects of the class to which the attribute belongs and objects of classes that inherit from the given class.

- Attributes are entered in the second section of the class symbol in the class diagram. Full specification of the attribute in the diagram (in the very extensive form, used in the implementation phase) should include the following elements (Wrycza et al., 2005):

```
<atribute> ::=[<visibility>]["/"][<attribute-name>][":"
<type>] ["<cardinality>"]["="<initial-value>]["{<determining-
ownership>"}],
```

At the conceptual stage, not all of the specification elements need to be used, just the following representation:

```
<atribute> ::= [<visibility>][<attribute-name>][":" <type>]
```

Services

P. Coad and E. Yourdon, (1994, p. 131) define a service (or operation or method) in the following way:

it is a specification of the behaviour, which it is bound to manifest according to the scope of the system.

The authors quoted above suggest questions that should be answered by an analyst when modelling services:

- What services does the object require to change its states (understood as the values of attributes)?
- What services are required by the objects of other classes (identification of interactions between classes)?

Types of services

The literature (Coad and Yourdon, 1994) proposed the following differentiation between services:

- Simple algorithmic services related to creating, modifying, and deleting the object and its attributes. The most commonly used algorithmic services:
  - Create
  - Connect – sets up (deletes) links to other objects
  - Get/Set – the service gets or sets the attribute values
  - Release – the service releases (disconnect and removes) an object
- Complex algorithmic services go beyond simple services described above and perform:
  - Calculations based on attribute values
  - Monitoring of the external data source

Technical services, such as Initiate and Finish, may be related to the latter type of services. One of the characteristics of services is predefined visibility for attributes.

Notation

In the class diagram, services are in the third segment of the class symbol. In UML notation, the specification of services includes the following elements:

```
Service::= visibility service-name (parameter-list): result-type-
expression {property string}
```

Messages

Messages are the method of using services. In order to perform a service of another object, the sender sends a message containing the name and other parameters of the service defined by the executing object. P. Coad and E. Yourdon (1994) extend this concept by stating that the links corresponding to the messages model the dependencies of the object on the processing process, demonstrating the need to use the services to complete the sentence.

Relations

The most important relations, which may include classes in the diagram discussed, are:

- Associations
- Generalisations
- Aggregations

Different types of relations are presented below.

Associations

An association is a relationship between two or more classifiers (in our case, classes), describing the relationship between their instances (Wrycza et al., 2005).

Another definition states that associations in the class model define the relation model in the problem domain, in which the object must be with other objects in order to fulfil its obligations (Coad and Yourdon, 1994).

Notation

Associations are shown in the class diagram (Fig. 2.32) as the lines connecting the class symbols. Associations can have names, which express their meaning and significance in the problem domain. The names are placed above the line showing the association. UML notation also permits indication of the direction of the interpretation of the association by placing an arrow mark next to the name of the association. Directions of the interpretation of the association do not mean one-way communication between the connected classes.

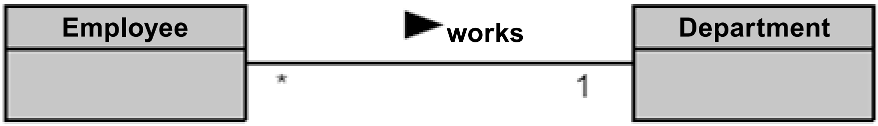

Fig. 2.32 Diagram of the Drawing classes in named and bidirectional association
Source: Own work

Cardinality

Cardinality helps to determine how many objects at each side of the association may participate in a single relation.

Notation

** - indicates 0 or more objects*
1 - means exactly one object
0..1 - means zero or one

The association shown in Fig. 2.33 will read as follows: *The customer can place a few orders, but each order can belong to one customer only.*

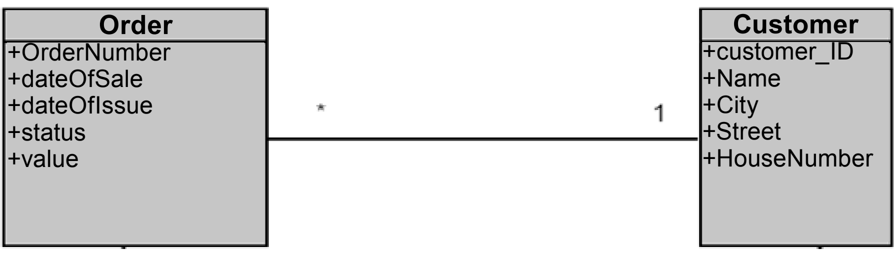

Fig. 2.33 Class diagram – associations with cardinality
Source: Own work

Aggregation

Aggregation is another type of relation that occurs between classes. This relation reflects the whole-part relationship, which is one of three "object-oriented" methods of organisation. There are many examples in reality – a car and its parts – engine, wheels, etc. When naming this type of relation, the verbs "has" or "is made up of" are mostly used.

UML defines two kinds of aggregation:

1. Total aggregation – a composition
2. Partial aggregations

The differences in these types of aggregation consist in stability of this relation. In a total aggregation, parts may not constitute independent entities i.e., they exist only in connection with the object illustrating the whole, and all its components are also removed at the time of removal of the aggregate object. Total aggregation is shown in Fig. 2.34. It is the relation between the Order and the Order Item objects.

Generalisation

A generalisation, described by some authors as a gen-spec structure is the grouping of objects belonging to a common group, and differing from each other by a specific number of type-specific attributes or operations.

The notation of generalisation is a line with an empty triangular arrow pointing to the base class or in other words the generalised class. It is worth mentioning that in the derived class, the components inherited from the base class are not referred to in the diagram, even though they are physically present there. An example of a generalisation is the base class Customer with the subordinate classes Individual Customer and Corporate Customer the following diagram (Fig. 2.34).

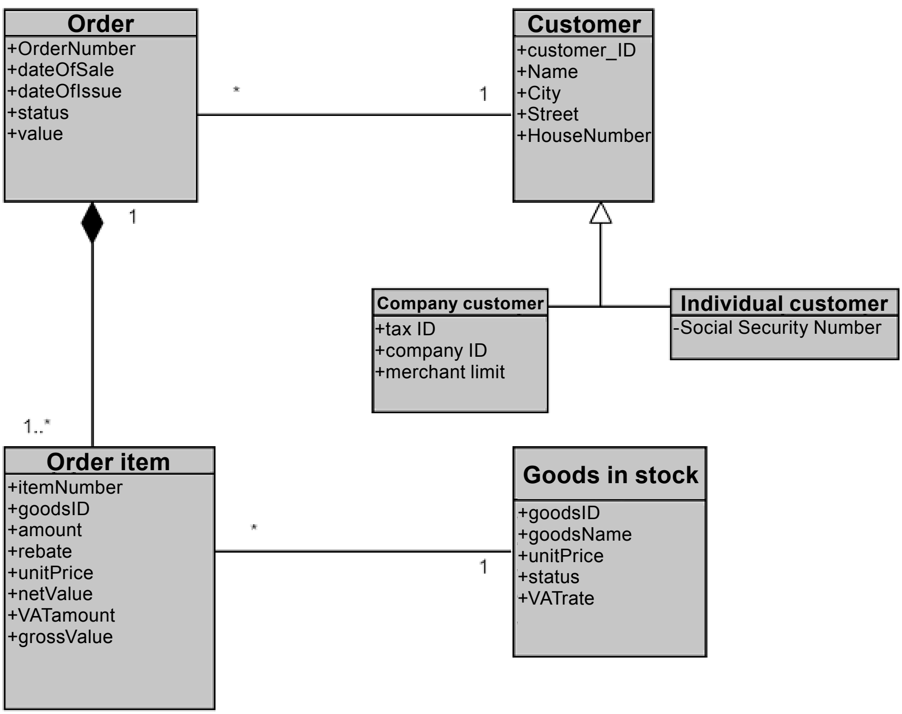

Fig. 2.34 Diagram of a trading system classes
Source: Own work

Use case diagrams

The technique of describing user requirements by describing the interaction between the user and the system occurs in many analysis and design methods, even those “pre-object-oriented”.

Previously this was done in a very undefined and non-formal way. In object-oriented methodologies, the use case played a significant role. For the first time, this concept was defined by Ivar Jacobson, who formally described the process of documentation of use cases in his OOSE methodology. Ivar Jacobson was one of the “three musketeers”, who created the unified methodology and the UML standard. Use cases are now an extremely important method for describing functional requirements of the system.

Basic concepts

In order to define the concept of a use case, we must refine the concept of a scenario. A scenario is a sequence of steps describing the interaction between the user and the system. When describing the subsequent steps of the scenario, their different variants can happen, as well as other conditional alternative paths of the basic scenario.

M. Fowler and K. Scott (2002) define a system use case as:

a set of scenarios related by a common goal.

More or less formalised scenarios are the basic language to describe use cases. UML does not precisely define the specification standard adopted. But it proposes use case diagrams, which show a set of use cases that occur in the modelled system or part thereof. Use case diagrams belong to the group of dynamic diagrams modelling functions implemented by the system.

Notation

Diagram notation for the system use cases includes: actors, use cases and relationships between them. An actor is a role that the user plays in relation to the system. Actors can include any elements from outside the system, which will contact the system. Some authors have divided actors into human and non-human actors (Wrycza et al., 2005). Human actors include people, groups of people or organisations interacting with the analysed system. Non-human actors are external systems or devices. Time can also be an actor in the use case. An actor may implement one or several use cases. Graphical symbols used in the use case diagram are shown in Fig. 2.35.

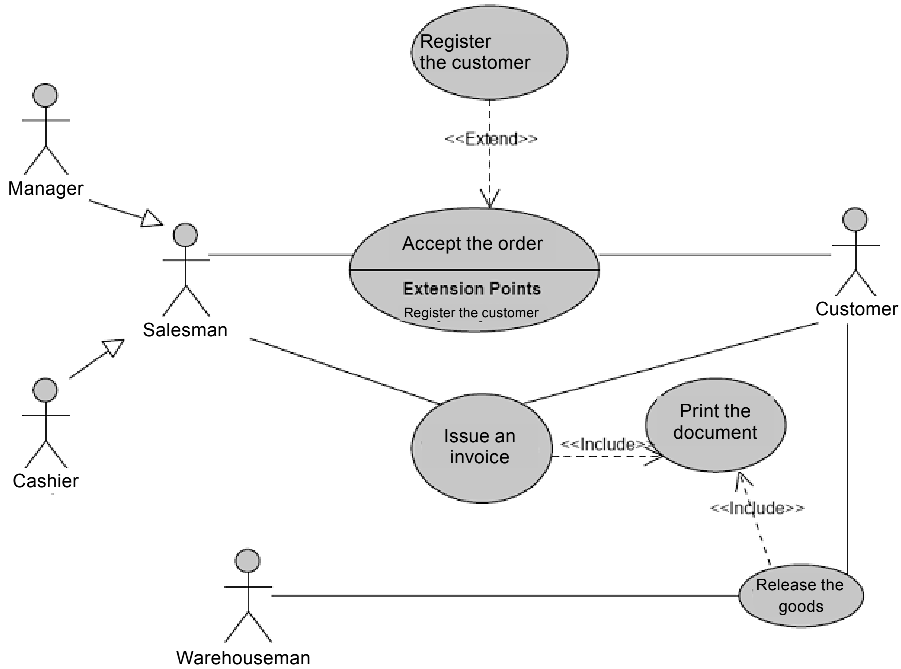

Fig. 2.35 Use case diagram for a trading system
Source: Own work

The names of actors are usually nouns or nouns with adjectives, always written in the singular. Use case names should be semantically related to the purpose of an actor. They are written as verbs with objects. Whereas verbs are used in the imperative mood: take an order, valuate the contract, determine the limit.

The line that shows the association between the actor and the use case represents the interaction taking place, and not the flow of data – as is the case with data flow diagrams (described in the chapter on structural methodologies).

Associations (semantic relationships) between actors and use cases in the diagram do not have names. Optionally, cardinality and navigation can be shown.

Inclusions and extensions

Use cases may enter into inclusion relations with each other labelled with a dotted arrow with the stereotype <<include>> and extension relations indicated by a similar arrow with the stereotype <<extend>>.

Inclusion

Inclusion occurs when several use cases have the same set of steps of the scenario, and they can be included in a separate use case. Inclusion is mandatory, i.e. both use cases must be implemented. Another reason for using the inclusion is if the use case scenario is very large and it should be divided for clarity. See the <<include>> relations in Figure 19.

Extension

Extension is a relation, in which the extended (basic, or base) case makes optional steps of the extension case, but after the fulfilment of appropriate additional conditions. Extension occurs only in certain situations named extension points. An extension is an optional relation. The most practical use of extension is the occurrence of multiple asynchronous services, which the user can use during the basic use case (Fowler and Scott 2002). Extension points are placed in the middle of the symbol of the use case or in the appropriate place in the text scenario of the given case. An extension point indicates a situation where the extension case may be called. See the previous figure. See the <<extend>> relations in Figure 19.

Use case documentation

A system use case diagram shows what kind of actors are involved in processing, what situations in the life of the organisation the system will handle and the relationships between use cases. A use case diagram is a kind of map of system functions. The diagram does not show in which steps the given interactions with the system will be implemented.

UML does not define the method of describing the scenario and the other elements of the use case.

The literature describes a number of ways of documenting use cases:

- Natural language, non-formal system
- Formatted templates
- Tables
- Pseudocode
- Activity diagrams

In the book *Write Effective Use Cases* by A. Cockburn (2004), which is a compendium of knowledge on the subject of use case description, the author has analysed different types of specifications and has not indicated the only and the best one. The choice of the description method depends primarily on the purpose, scope and stage of the design work. For the purpose of this textbook, the one preferred by Cockburn is presented – the template of a fully formalised use case (Table 2.7).

Table 2.7 Template of a fully formalised use case

Use case No., template of a fully formalised use case <name>
<the name should be the objective in the form of a short active verb phrase>
Context of use: <longer description of the objective; if normal conditions for occurrence are required>
Scope: <design scope; which system is treated as the design black box>
Level: <one of: summary, user objective, subfunction>
Stakeholders and Interests: <list of stakeholders and their key interests in the use case>
Initial condition: <what state of the world do we expect at the beginning>
Minimum guarantee: <how are the interests protected at any ending>
Success guarantee: <state of the world, when the objective will be achieved>
Trigger: <what triggers the use case, can be a time event>
Main success scenario: <place the steps of the scenario here, from the trigger to the objective achievement and the later clean-up> <step number> <action description>
Extensions: <place the extensions here, one per row; each one refers to a step of the main scenario>
<altered step><condition>: <action or inferior use case>
List of options for technology and data: <place the options, which can conclusively cause a division of the scenario> <step or option number> <list of options>
Additional information: <anything needed in the project as additional information>

Source: Own work based on (Cockburn, 2004)

To read about other forms of documentation of the use cases it is recommended to refer to the book (Cockburn, 2004).

Business vs. system use cases

Description of the requirements for an information system in business organisations at the conceptual stage requires an examination of the organisation in a broader perspective than planning and specifying the interactions with the IT tool. One should begin with a general overview of how the organisation works, with what external elements it is in contact, and thus the presentation of the model of business processes.

The perspective of business processes enables to show the directions of the reorganisation of processes in order to improve efficiency. Another important element of this approach is better communication with business users, who pursue business processes on a daily basis – information processes are only one of the tools used every day.

To illustrate the business processes using UML, a clear separation between the business use case and the system use case has been created. System use cases, which we referred to up to now, show the user's interaction with the system to achieve the objectives of the user. In business use cases, an enterprise is modelled instead of a system, therefore the business use case shows the interaction of the business user with the modelled enterprise.

Since the topic of business modelling is a very current, a number of graphics-oriented stereotypes were created, focused on showing business classifiers to distinguish them from a system perspective.

The business use case diagram shown below is a reflection of the situation from Fig. 2.35, where system cases for a trading system are shown. At a more general business level, this can be expressed as the business use case Sell Goods, where the business actor Customer contacts a company to purchase goods. In this case, the use case presents not a system, by the entire modelled organisation.

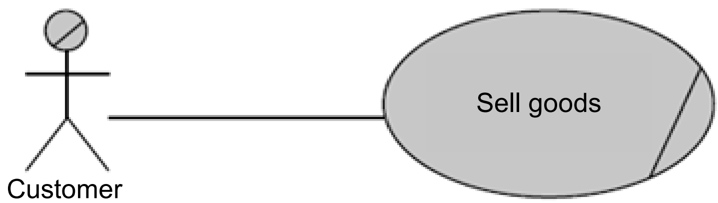

Fig. 2.36 Business use case diagram
Source: Own work

Other UML diagrams

It is not possible to describe all diagrams in this textbook. Only the most commonly used ones are presented. In this chapter, two diagrams belonging to the group of dynamic diagrams are shown in a very simplified version – the state diagram and the sequence diagram. These diagrams together with the class diagram and the use case diagram constitute the basic set and, as practice shows, are the most commonly used. Detailed descriptions of the other UML diagrams can be found in (Wrycza et al., 2005, Fowler and Scott, 2002) or at the source at www.omg.org.

The state diagram

A state diagram (or a state machine diagram) is used to show all the states in which the object can be found in its life cycle. A state diagram always refers to one class. The definition and notation of the basic elements is provided by (Wrycza et al., 2005), citing the OMG standard:

- *State* – a circumstance or situation in which the object is during the life cycle.
- *Transition* – a relation between two states indicating that the object in the first state will perform certain actions and move to another state when a specific event will take place and specified conditions will be complied with.

- *Initial state* – initialisation of the state machine.
- *Final state* – the end of the state machine.

The following example (Fig. 2.37) shows the life cycle of one of the important objects of the trading system, namely the Order object.

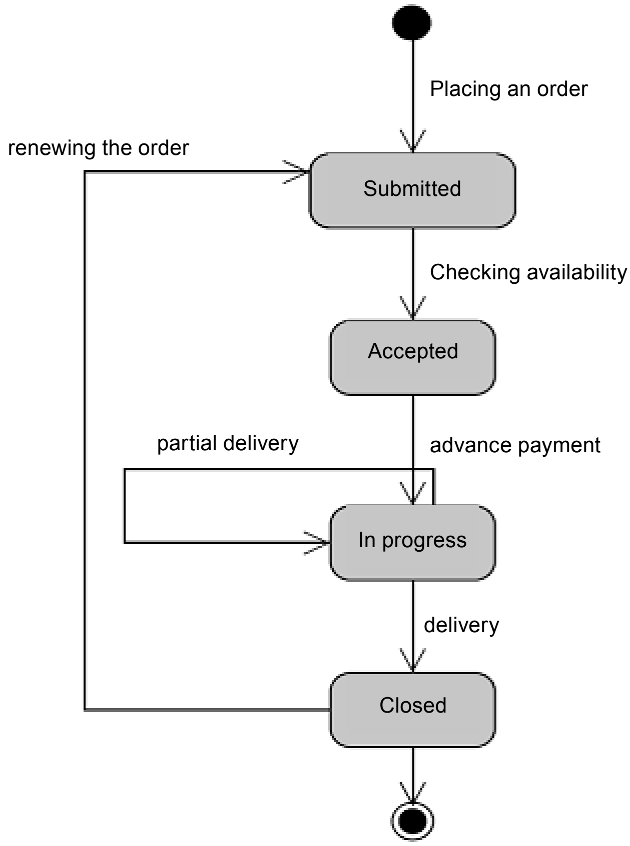

Fig. 2.37 State diagram of the Order object for the trading system
Source: Own work

A state diagram is also shown in the design of a sample organisation in the next section.

The sequence diagram

A sequence diagrams belongs to the group interaction diagrams. Interaction diagrams show the interaction of groups of objects as a part of a use case (Fowler and Scott 2002).

Notation

Object – One of the participants in the processing sequence. Rectangle notation.

Life line – Represents the active time of the object in the processing sequence. A line below the rectangle that describes the object.

Message – Shows the transfer of a query (usually running the appropriate service) between objects. An arrow between the life lines of objects.

Fig. 2.38 shows the sequence diagram for the use case derived from the trading system – Accept an Order.

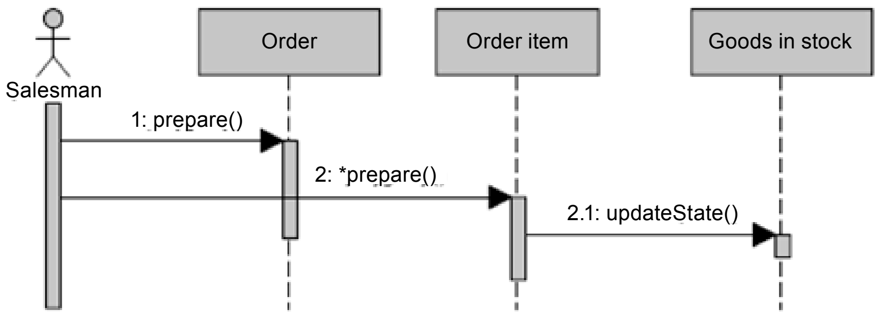

Fig. 2.38 Sequence Diagram for a use case – Accept an Order in the trading system
Source: Own work

2.3.3.4. RUP methodology as a software development process

UML is a collection of diagrams and techniques that enable the modelling of information systems, however it is not a software development methodology. The methodology provides a recipe for what methods and tools and should be used and when in order to achieve the objective of the project. This section describes the basic phases and tools of the RUP (Rational Unified Process) methodology, which, because of the common roots with the UML language, originated at Rational Software, and in a natural way uses UML in the software development life cycle.

The basic assumptions of the RUP methodology are the iterative and incremental software development cycle and the use of graphical models in the whole cycle. Other assumptions include: requirements management, use of component-based architecture and strict quality control.

The iterative and incremental software development process consists of cyclical supply of parts of the system for testing and acceptance by the user. In the classical waterfall approach, testing took place after the implementation of the entire solution, and only then errors were picked up, sometimes resulting from the initial stages of the cycle, such as the analysis. The costs of such amendments are dramatically higher. The iterative and incremental approach provides the subsequent versions of the system at a given time called iteration. At each iteration, the new version of the system is created, tested and accepted. Each iteration includes a system development cycle with basic steps: analysis, design and programming. Actions performed in different iterations are named disciplines in the RUP methodology. At each iteration, all of the following areas are present, but with varying intensity:

- Business modelling
- Specification of requirements
- Analysis and design
- Programming
- Testing
- Implementation
- Configuration and change management
- Project management
- Preparation of the environment

In the context of the project phases, the RUP methodology introduces four phases:

1. Inception
2. Elaboration
3. Construction
4. Transition

The phases (especially construction) internally consist of successive iterations, after which "incremental" versions of the newly emerging system are available.

Characteristics of phases with an indication of the role of UML in a RUP methodology cycle:

- *Inception* – in this phase, high-level characteristics are developed, the economic and time desirability of the project is assessed. Based on these elements, a decision is made about starting the development phase (or project completion). In the context of UML tools, a context diagram (an object-oriented version, based on use case notation) may appear at this stage.
- *Elaboration* – the task of this phase is to prepare the specifications of the target solution. The scope of the project is defined in this phase. Use case diagrams showing the functions, which the target system should implement are used for this purpose. It is also necessary to become familiar with the conceptual dictionary of the analysed domain with the class diagram being the most important element here. Then, the services of particular classes are specified, for which interaction diagrams and state diagrams are necessarily, if some objects have a complex life cycle. For enterprises

engaged in complex business processes, elaboration can start from an activity diagram or a business use case diagram.

- *Planning for the construction phase* – is based on a selection of the sequence of iterations, and therefore the use cases implemented in their course. This approach gives a clear arrangement of the design products, i.e. the functions provided at the end of the given iteration. An important element is the appropriate arrangement of the implemented use cases, in order to avoid a deadlock or inability to provide one of the implementation paths, due to the lack of functions from a different path, in which the needed use case is scheduled for the subsequent iteration. On the basis of a complete iteration plan, programmers can begin the process of determining how time-consuming the implementation of the various use cases is. Based on estimates of time consumption, the length of the iteration is determined. To maintain the rhythm of the project, the iterations should have an equal length (measured in man-weeks), typically from two to eight weeks (Fowler and Scott, 2002).
- *Construction* – In this phase we use all UML diagrams created in the elaboration phase. Some of these require refinement, since now the project is already in an implementation phase (and thus requires the highest details). The primary work of the programmer during the implementation phase is to verify classes designed with the needs of the use cases, which he is just starting to implement. Interaction diagrams and CRC (Class Responsibility Collaboration) cards, as described in (Fowler and Scott, 2002), are suitable for this purpose. With these tools, we verify the completeness of the predefined class services and attributes. At the same time, we can complement them with implementation elements (unnecessary in previous phases). The use of the UML language does not end with the delivery of the system. The documentation developed should be used and cultivated also in the operation phase of the system. In the projects of large systems, package diagrams also play an important role, by which we can get to know the logic and modules of the created solution. The so-called test use cases are very useful for incremental and final testing. Test scenarios are designed on the basis of use cases in order to manipulate specific input data and correctly calculated output values. The correct functioning of the system can be unambiguously determined based on appropriate tests performed by the user (UAT).
- *Implementation* – in the last phase, the system is handed over to the user. But before this occurs, the appropriate hardware and software configuration must be prepared. For this, UML provides two graphical tools, the deployment diagram and the component diagram. The component diagram shows the relationships between system components. The deployment

diagram shows the physical ICT structure of the implementing organisation and the target location of the components of the new system.

2.3.3.5. The design of an information system for a sample organisation using the object-oriented methodology (selected UML diagrams)

This section shows fragments (Fig. 2.39 - Fig. 2.42) of the IS design for the *Cool Film* rental shop, similarly to the example of the structural method. As a part of the object-oriented design, the use case diagram is shown that is compatible with the list of events shown in the environmental model of structural methodology. Then, the class diagram is shown, the state diagram for the film object, and the interaction diagram for the Rent a Film use case.

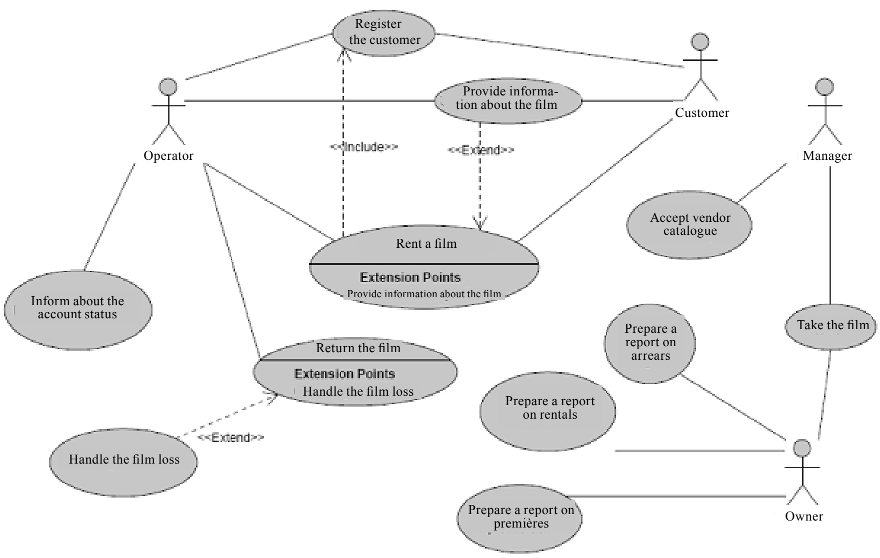

Fig. 2.39 Use case diagram for the Cool Film rental shop system

Source: Own work

Fig. 2.39 shows a use case diagram created from the list of events, discussed in greater detail in Section 2.3.2.7.

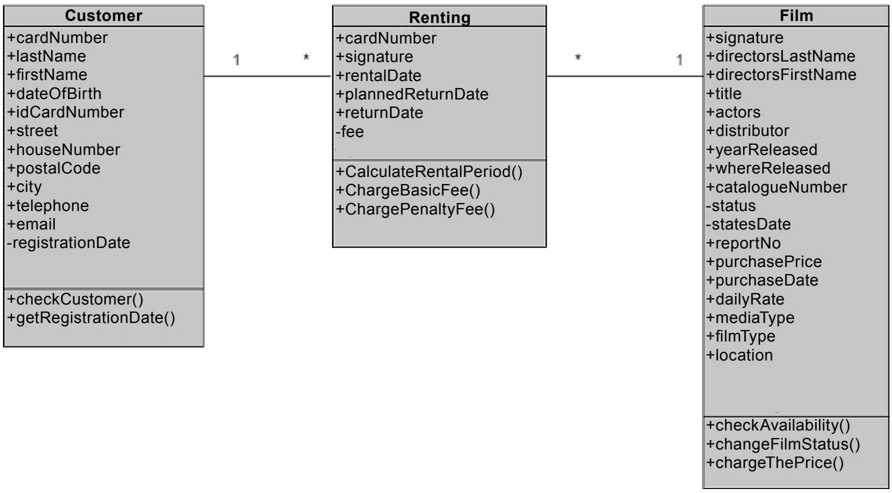

Fig. 2.40 Class diagram for the Cool Film rental shop system

Source: Own work

A class diagram (Fig. 2.40) shows the most important relation in the system of an organisation such as the rental shop, meaning the Customer Rents a Film relation.

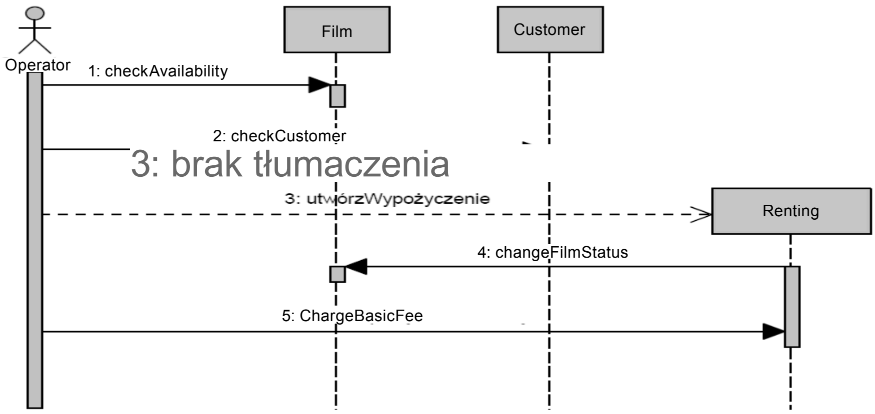

Fig. 2.41 Sequence diagram for the Rent a Film use case in the Cool Film rental shop system

Source: Own work

Fig. 2.41 shows the interaction between the actor and the classes involved in the Rent a Film use case.

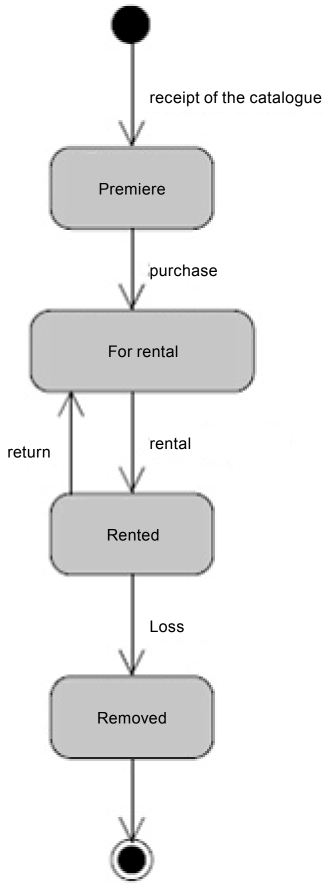

Fig. 2.42 State diagram for the object Film for the Cool Film rental shop system
Source: Own work

Fig. 2.42 shows the states through which the Film object passes during the life cycle.

2.3.3.6. IT tools supporting IS analysis and design. The definition and evolution of CASE tools.

The introduction of support software engineering tools called CASE tools to the set of tools of the computer analyst had a great impact on the evolution of methods of structural analysis, and those subsequently developed. After the structural declaration of the use of graphical tools (data flow diagrams, entity relationship diagrams) in the analysis and design, the labour-intensity of their creation and subsequent care using "paper and pencil" methods began to be noticed. Manual processing of diagrams, in particular for large systems, was vulnerable to errors and inevitable outdatedness. Therefore, with the advent of microcomputers and programs for processing text and images, their widespread use in analytical work began. Initially, these were only tools able to "draw nicely". The next step was the introduction of specialized packages able not only to draw a specific diagram, but its validation in accordance with the implemented methodology. Initially, CASE tools allowed drawing individual diagrams, in time more sophisticated tools appeared that supported diagrammatic tool sets defined by the analysis and design methodology. In the eighties, the structural paradigm was most popular, but in a few notational variants. The lack of a single approach significantly hindered the creation of CASE tools. Developers had to decide which notation and methodology their tools should support. A very important step in the evolution of CASE tools took place with the introduction of the object-oriented paradigm, and the UML language with it, which had been unified and adopted as a standard. From this point on, CASE tools are evolving according to the development and subsequent versions of UML.

Definitions of CASE tools

S. Wrycza (1999, p. 178) provides the following definition of CASE tools:

> *Tools supporting the automated creation of IT systems throughout the life cycle of the system. CASE packages are applied in computer technology to processes, techniques and methodologies for IT system development.*

Features of CASE tools

The basic features of CASE tools include:

- The capability of using graphical (and text) tools for IS modelling.
- The capability of semantic control on the level of each graphical and text tool. This feature distinguishes CASE tools from purely graphical tools. CASE tools have built-in "expert" knowledge about the principles and methods of analysis and design.
- The capability to control the compatibility between different graphical and text models within the entire project such as checking compliance of DFD diagrams with ERD in structural methodologies.
- Support for the project repository (sometimes called a project dictionary). The repository stores the descriptions and characteristics of the elements used in all models related to the project. This element is critical, especially in large projects carried out by several work groups.
- Support for group work.
- Automated generation of program code or database structures.

- Automatic generation of reports. Paper documentation is created in analytical and design work, usually in large amounts. Therefore, machines that generate reports in certain configurations save a lot of the project team's time.
- The capability of testing and simulation of modelled system components.
- Possibility of multiple use of the modelled of system part in various projects (creating own templates).
- Support for the creation of prototypes.
- Support for reverse engineering – which allows CASE tools to import the finished parts of the database structure or program code in order to modify them.

Typology of CASE tools

There are several criteria for the division of the above tools in the literature; we will describe three of them in this textbook (Wrycza, 1999):

1. Comprehensiveness
2. Methodological compatibility
3. Support for the stages in the life cycle of IS

Comprehensiveness means the number of modelling tools contained in the described CASE tool. This criterion is divided into the following categories:

- *Partial* – support only selected single text or graphical tools are related to the particular methodology.
- *Indirect* – support several analytical tools in the selected methodology.
- *Integrated* – support the full life cycle of creating the IS usually assigned to a particular methodology, although there are also tools supporting several methodologies (for example the Visual Paradigm package, which will be described in the next section).

The criterion of methodological compatibility is based on the support for structural, object-oriented and social approach. This criterion also includes tools supporting several methodologies.

The criterion for the support of the stages in the life cycle of IS lists the following categories in the described tools (Wrycza, 1999):

- Strategic and economic modelling planning (earliest stages)
- High level stages (*Upper CASE*)
- Low level stages (*Lower CASE*)

In the above criterion, some authors also list the integrated package, which can support the full cycle of IS creation.

Case Studio – as a tool supporting the structural methodology

Case Studio developed by *Chronoware*, currently *Quest Software*, is a dedicated tool for the structural methodology. In the context of complexity, it is an intermediate class tool – it does not implement all structural analysis methods. It offers support in building data flow diagrams (DFD) and entity relationship diagrams (ERD). All examples of diagrams in the chapters describing the structural methodology were created with this tool. The main disadvantage of this tool is the lack of a central project repository. *Case Studio* offers generation of database structures directly from ERD diagrams. It also supports reverse engineering. In the context of the supported IS life cycle, this tool can be used in analysis and thanks to functions generating data structures, in the design and implementation stages, but only in the database area. Therefore it is a tool in the *Upper Case* category.

Visual Paradigm – as a tool supporting the object-oriented methodology

Visual Paradigm is an integrated package that can support the work in all stages of IS development. Tools featured in this package are grouped into the following categories:

- UML (standard 2.0 compliant) – 15 diagrams.
- Process modelling – BPMN, DFD, EPC notation – 6 diagrams.
- Database modelling – ERD and ORM – 2 diagrams.
- Requirements modelling – SysML diagrams and CRC cards; this category also includes tools for text analysis in regard to the search for analytical objects.
- Remaining tools support user interface design in regard to forms and reports. Visual Paradigm also supports popular communication technology (based on graphical tools) used in analytical meetings, the so-called mind mapping.

Such a set of tools confirms that Visual Paradigm is an integrated suite for all categories mentioned earlier. The product also features a customisable report generator and imporing models from most well-known analytical tools. Developer versions also allow to integrate the tools with programming platforms: *Eclipse*, *Visual Studio.net*, or *Netbeans*. All the graphical examples in the chapter on object modelling were prepared in Visual Paradigm, the Agillian version.

2.4. SELECTED PROJECT MANAGEMENT TOPICS

2.4.1. Project management methodologies

Effective project management requires a properly planned approach that will ensure a holistic recognition of all the project conditions. It is caused by the diversity of projects, a multiplicity of areas which project managers have to deal with, as well as the variety of stakeholders who are involved in the process of project management. The most popular project management methodologies include PRINCE2 and the methodology proposed by the Project Management Institute as defined in the PMBOK (Project Management Book of Knowledge).

The variability of the environment and the requirements of certain types of projects, such as Internet applications and research and development projects aimed at developing new products, led to the creation of extreme project management. This approach assumes that the success of the project is decided by the ability to make quickly introduced changes during the project and to deal with difficult and complex situations. Extreme approach to project management is referred to as "agile project management" and uses the so-called agile project management methodologies, which include SCRUM, DSDM (Dynamic Systems Development Method), FDD (Feature Driven Development) and Crystal Light (Wysocki and McGary, 2005).

2.4.1.1. Prince-2 methodology as an example of the disciplined project management methodology

PRINCE2 stands for PRojects IN Controlled Environments. It is a project management methodology developed during the long process sponsored by the UK Government and is available for free. PRINCE2 is a methodology used in many industries, it includes mechanisms for defining contributing partial products of projects and ensuring their delivery on time, within a given budget and in accordance with established customer quality requirements (Kerzner, 2005).

PRINCE2 is an example of a disciplined methodology, in which components, processes and roles of participants are clearly separated (The Office of Government Commerce, 2009). This methodology defines 40 actions, which are organised into seven main processes:

1. Starting up a project
2. Managing a project
3. Initiating a project

4. Controlling a stage
5. Managing product delivery
6. Managing stage boundaries
7. Closing a project

The methodology clearly emphasises a number of components (subjects), thus providing a tool for a comprehensive and holistic project management. Separate perspectives include business motivation, organisation, quality, plans, risk, changes, and progress. PRINCE2 also places emphasis on providing the multiple stakeholders perspective in the project and distinguishes various roles: Project Council, Primary User, the Board, Chief Supplier, Project Manager, Team Leader, Project Security, Project Support.

2.4.1.2. SCRUM as an example of the agile project management methodology

SCRUM is an example of the agile project management methodology and has its source in software projects. In the context of software projects, like other agile methodologies, SCRUM puts more emphasis on individuals and the interaction between the participants, operating partial versions of the program, collaboration with customers and responding to changes (Boehm, 2002). SCRUM methodology approach is to manage complex projects using iterative and incremental approach to optimise and control risk. The basic assumptions of the methodology include transparency, frequent reviewing and adaptation (The Scrum Guide, 2011).

SCRUM methodology distinguishes several specific roles, which include:

- SCRUM Master – leading the project and ensuring the observance of the rules and acting in accordance with the rules, can be treated as equivalent to the project manager.
- Product owner – responsible for the structure of the product and maintaining its subsequent versions.
- Team – a group of people responsible for delivering the product and its various versions, having various skills and experience.

Like other agile methodologies, SCRUM does not define the entire project at the beginning, but is based on an empirical approach, in which the whole project is divided into time segments calling them sprints, lasting usually from one week to four weeks and aiming to deliver the next version of the product. The segments occur without the intervals between successive editions, each sprint starts with a meeting intended to estimate the previous edition of the sprint, followed by scheduling the current edition, which in turn ends with a meeting intended to review the results of the current version of the sprint.

2.4.2. Change management

The implementation of projects related to information systems involves the organisational transformation of the enterprise and is related to change management. Change management is a multi-step process by which managers should primarily focus on the proper preparation and conduct of employees through the entire operation, making them aware of both opportunities and risks associated with the changes. The change management process comprises the following steps (Clarke, 1997):

- *Analysis of conditions for changes* – the purpose of this stage is to examine the attitudes of employees to the planned changes and to create a positive atmosphere for a change. This is done through the involvement of employees in the change process that consists of the earliest possible inclusion of employees in diagnosing the problems of the company and bringing about their own determination of reasons for changes.
- *Anticipation of resistance* – at this stage, the resistance of people against the planned changes is estimated, which may be associated with the modification of the scope of changes. The risk of changes and the level of organisational readiness are determined. The result is an initial indication of the extent of the changes and assessing their potential impact on employees and customers.
- *Building a shared vision* – this stage consists of communicating the vision of changes to employees and strengthening the sense of purpose. At this stage, conviction should be built among employees that the benefits of the process of change outweigh the costs.
- *Getting people involved* – this stage should focus on expanding the group of persons involved in the process of changes. For this purpose, project groups are set up and change agents and moderators engaged. Consultations are held and there is extensive communication with individual employees and groups.
- *Creating a plan for change* – at this stage, a phased plan of changes is created, continuing over time. The schedule of changes should include the stages possible to be implemented and ensure the required prerequisites. There should be formal bilateral channels of communication between change managers and employees.
- *Strengthening the changes* – this step consists of the implementation of the plan of changes and ensuring their sustainability. Changes should be "frozen" and institutionalised by adapting appropriate procedures and systems. Care should be taken to ensure an internal climate which continuously promotes changes and no turning back to the previous behaviour.

2.4.3. IT tools supporting project management

Project management is a vast area that includes a variety of projects involving different resources. Projects may take the form either of short projects, which involve only a few people, as well as global projects involving hundreds of people from many countries. Effective project management thus requires the support of appropriate software, whose functionality should match the type of the project.

One of the most widely used project management tools is *Microsoft Project*, which aims to support the project manager in developing the project plan, allocating resources, tracking progress and managing the budget of the project. Microsoft Project uses the critical path analysis method (CPM) and allows for visualisation of the project plan and allocated resources in the form of a Gantt chart. It also provides a capability to define different categories of users with varying levels of access to project data and documents.

Of the many available programs supporting project management, worth mentioning is the tool called *ProjectPlace*[35], which allows for continuous project management using the SaaS model (software as a service). The functionality of the tool consists of supporting the activities of a distributed project team whose members can have instant access to the program from anywhere via a Web browser, including through the use of smartphones. The program stores the design data and documentation centrally on a remote server, contains a range of functions that support teamwork, such as remote conversations, document review and online meetings.

Also worth mentioning are *Basecamp*[36] by *37signals*[37], which can be rather considered a tool to manage teamwork, not a standard tool for project management. This software places emphasis on communication and cooperation between those involved in the project. Using *Basecamp*, project participants can exchange messages, post comments and reviews of documents, share files, assign and settle responsibilities, and reuse data from existing projects as models for new ones.

2.5. MODERN TRENDS IN MIS DESIGN

A typical information systems design area is to create subsequent, increasingly sophisticated management systems. Currently, these are mainly systems that

[35] http://www.projectplace.com/

[36] http://basecamphq.com/

[37] http://37signals.com/

might be categorised as DSS (Decision Support Systems), discussed briefly in Section 1.4.4, which currently are the most popular tools for computer-aided decision making in business. Simplified diagram of this system is shown in Fig. 2.43.

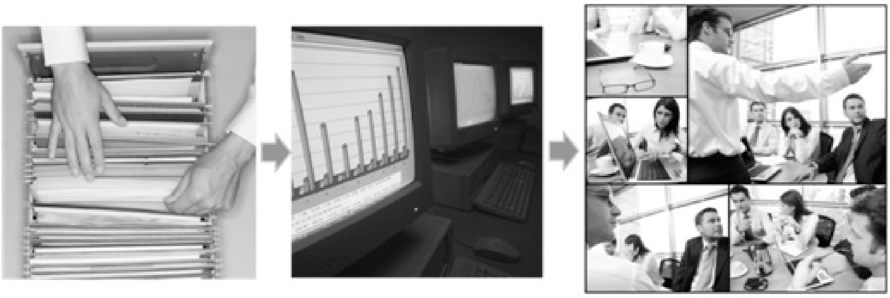

Fig. 2.43 Diagram of a typical operation using a DSS-system for management process support
Source: Own work

Today, the development of such systems is a task most needed in terms of user needs, because their design is the most common activity, which various IT teams are engaged in. Details of designing such a system and project management are already discussed in sections of this chapter.

This section is somewhat futuristic in nature. Not wanting to duplicate the content of other studies in this collective work, the author decided to present the results of a number of deliberations and evaluation studies here, related to the new generation of IT systems, whose operating principle is based on the achievements of cognitive science – the study of the cognitive processes, particularly the acquisition, processing and use *of knowledge* (Szymanik and Zajenkowski, 2004). Based on cognitive foundations, a concept of IT systems was built, which can penetrate into the semantic realm, that is, analyse not only the form of information (such as what words the text examined is composed of), but also what substantive meaning is contained in this information (Ogiela and Tadeusiewicz, 2009). Systems considered in general terms are defined as Understanding Based Cognitive Categorization Systems – UBCCS, but at the stage of detailed research, several subtypes of these systems are distinguished and examined separately (Fig. 2.44).

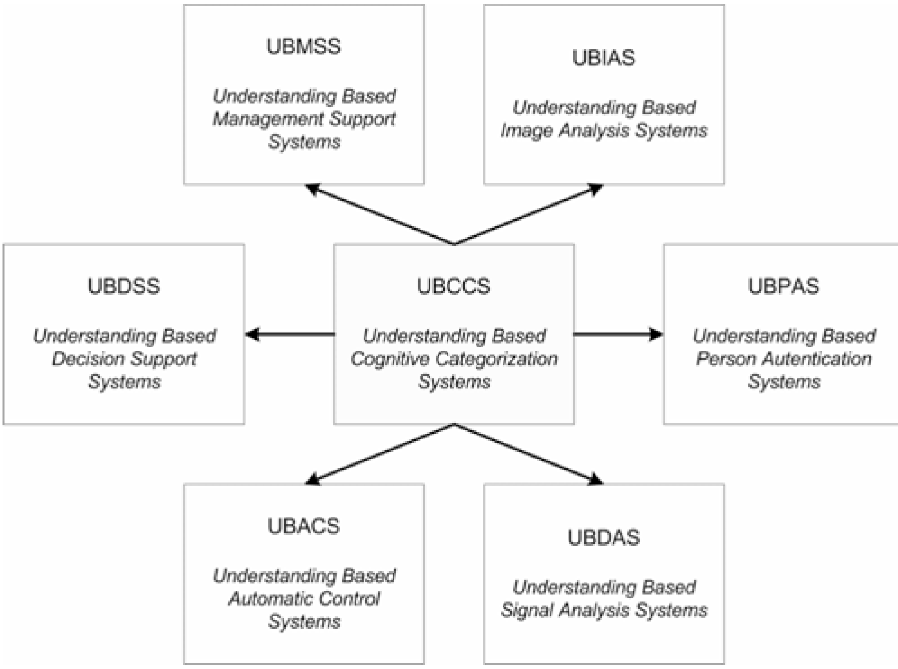

Fig. 2.44 Different types of UBCCS systems
Source: Own work

Such systems can be used in many areas. Their origin come from the work on automatic understanding of medical images (Tadeusiewicz and Ogiela, 2004), therefore the class of systems known as UBIAS (Fig. 2.44) is most often invoked in the literature on this subject. However, a growing number of more recent work refers to the UBMSS class of systems (see Fig. 2.44 and (Tadeusiewicz and Ogiela, 2008)) and these systems will be examined in this chapter.

Before proceeding to the presentation of details, a certain formal comment is necessary: Research on UBCCS class (especially UBIAS) systems is carried out by a research team, whose members are – in addition to the author of this section – also employees of the Faculty of Automatics at the AGH, Prof. Marek R. Ogiela and Mirosław Trzupek, MSc, as well as an employee of the Faculty of Management at the AGH, Dr Lida Ogiela. They are not listed as co-authors of the chapter, as this work is educational, not scientific in nature and has been prepared in response to a personal invitation sent by the editor of this textbook to the author of this chapter. However, it should be stressed that in terms of intellectual property, the concepts presented here (and illustrations) are the common property of the entire above-mentioned team of authors, often publishing scientific work together on topics presented here (see for example (Trzupek et al. 2009)).

2.5.1. Why the need for UBMSS class systems?

Let us examine any economic organisation (a company), and a typical system installed at this company (Fig. 2.45).

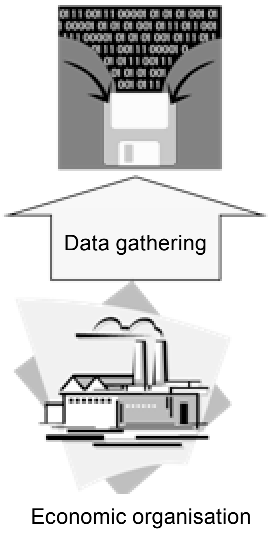

Fig. 2.45 IT system in a minimal configuration
Source: Own work

Such a system is in a minimal configuration and mainly performs record-keeping and billing functions, thus it collects data. However, there is little use of just collecting data. Therefore, better versions of the system feature analytical tools that will help to transform that data into useful information, and information can be used in management – especially on the lowest levels of the management structure, that is, at the operational and tactical level, when the tasks of the manager mainly amount to commanding various teams of employees (Fig. 2.46).

It is important to understand that full management of an enterprise also requires strategic management, but the person who is decision maker on the highest (strategic) level cannot rely on computer-aided decisions as effectively and as comfortably, as shown in Fig. 2.46. Strategic decision making requires something more than information alone (more or less thoroughly analysed). *Wisdom* is needed for strategic control, resulting from a deep *understanding* of the macro- and microeconomic situation, and that cannot be expected from modern computer systems.

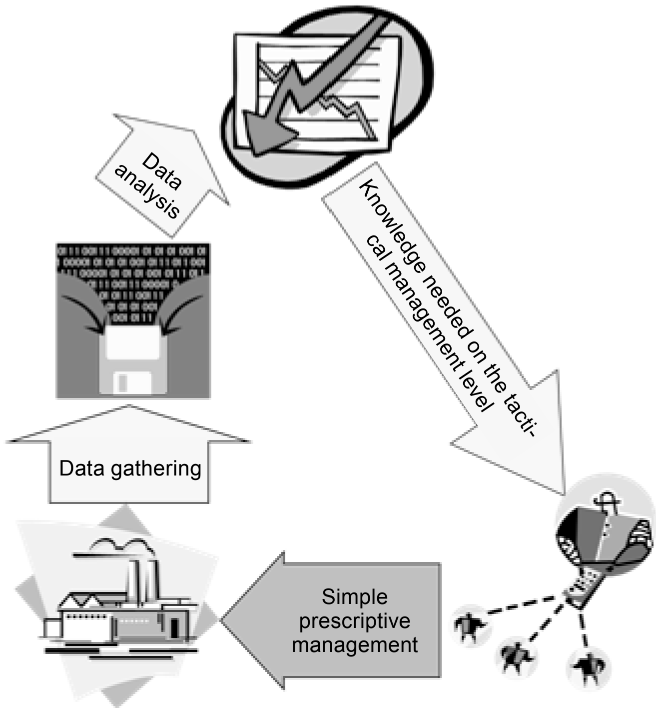

Fig. 2.46 Simple use of computer data analysis to support low level management (operational and tactical)
Source: Own work

Strategic decision-makers are obviously not deprived of basing decisions on the results of certain studies and analyses, but these studies and analyses are now made by teams of relevant experts, who are trying to properly *interpret* indicators derived from computer analysis, and on this basis they provide the decision maker with reports that are the basis for understanding the situation and taking an appropriate strategic decision. The diagram in Fig. 2.47 shows how it operates.

The challenge for developers of information systems in the near future will be to provide techniques for computer support also for the highest realm of strategic decision making. This is the purpose of the work on constructing the UBMSS class system.

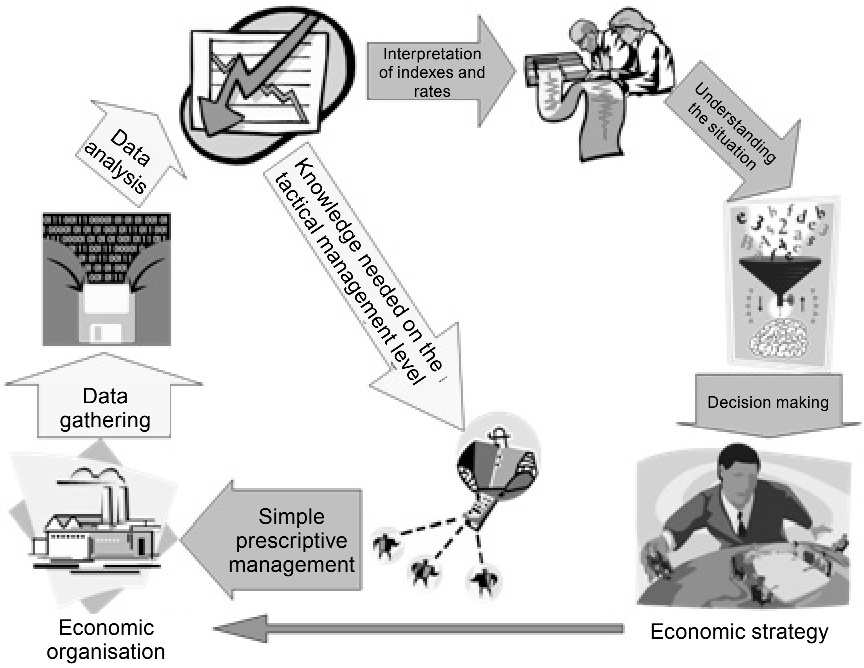

Fig. 2.47 Full management diagram, including both the tactical and strategic level
Source: Own work

2.5.2. How to achieve automatic understanding in management support systems?

Consider the diagram shown in Fig. 2.48. This diagram can be used to understand the messages of any kind, and therefore the ones that correspond to the understanding of economic data for strategic decision support.

In Fig. 2.48, two streams of information are visible, treated abstractly. One of them represents the current situation, so it is a stream of perception. The second represents the expert knowledge, therefore it is an internal interpretation generator for the perceived data. The collision of these two streams and their interference is the source (and the necessary condition) for understanding the meaning of messages flowing into the system.

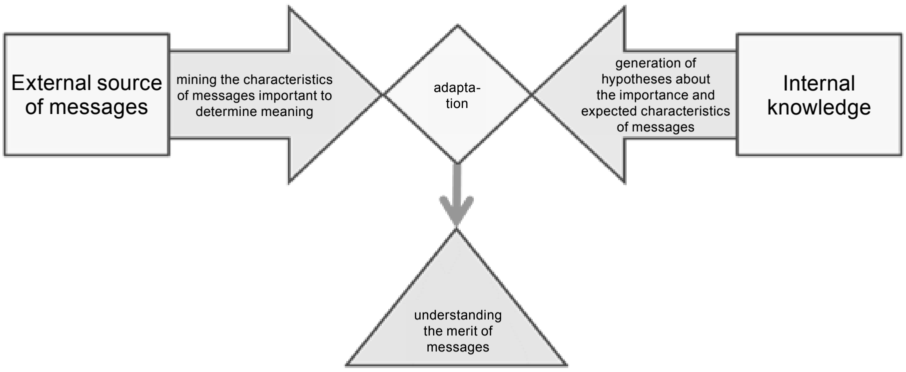

Fig. 2.48 General diagram of automatic understanding
Source: Own work

The role of knowledge in the understanding of the specific economic situations can be traced to two simple examples:

Let's first examine Fig. 2.49.

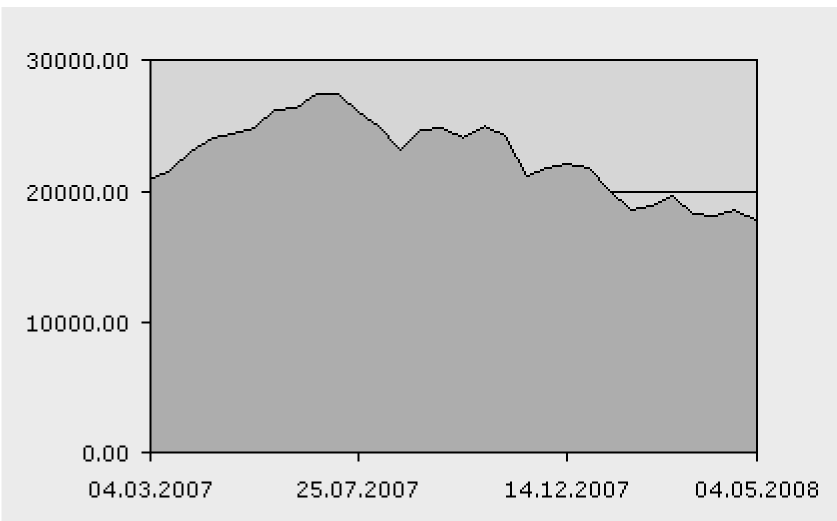

Fig. 2.49 Set of economic data, which can be easily interpreted in terms of its importance

Source: Own work

The drawing shows a chart, whose meaning apparently cannot be understood if there is no additional description. But if we look closely at this drawing, we see that the description of the horizontal axis looks like dates, and the description of the vertical axis could be associated with money. Combining these two observations with the knowledge that almost every reader has about the "turbulence" in the stock market between 2007 and 2008 – we can understand that we are looking at the decline in value of a stock investment. If we wanted to sum up this understanding in one short word, we could say: *a slump.*

Fig. 2.50 shows a very similar chart, which however can be the basis for a more sophisticated understanding of an event. We also identify this chart (on the basis of an analogy with the image analysed above) as an illustration of changes in the value of some asset listed on the stock exchange. However, in this chart, we can see a more interesting phenomenon. We can see, how after a long series of falls, the quotations began to slightly increase (around 6 December). Encouraged by this fact, the investor predicted that the upward trend started and bought a few thousand shares, which resulted in a sharp rise up the chart. Unfortunately, the hope for a rise (or even stabilisation of the rate) was deceptive – stocks instead of rising began to fall sharply and the entire payment was in fact lost. If we wanted to sum up this understanding of the meaning of the chart presented in one short word, we could say: *an investment error.*

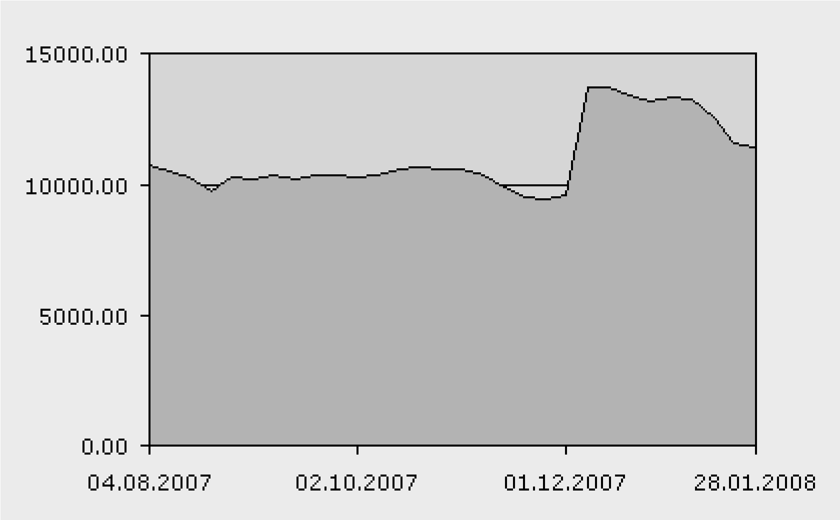

Fig. 2.50 Set of economic data, which can be the basis for the understanding of a certain event
Source: Own work

It is worth noting that this laconically summed up understanding of the situation consisted of two components: what can be seen in the set of economic data (shown for clarity in the form of a chart), and what the observer knows about the nature of processes on the stock exchange. Only the confrontation of these two sources of information gave rise to the understanding of the content carried by the analysed image. At the same time the fact that, based on the same image (and knowledge), we can reach to this substantive content – indicates that the image can contain specific content and that we can try to understand this content. And if man can understand this content – it is possible that computers can also, because over many years of the development of artificial intelligence (now called computational intelligence), it was repeatedly shown that whenever any form of human intellectual activity could be well defined and precisely described in psychology, and especially in cognitive science, then after a short time it was also often possible to build a computer program that could imitate this intellectual activity of man – and in many cases even exceed it.

2.5.3. UBMSS system construction

In order to start building the UBMSS system, we will transfer the elements from the diagram in Fig. 2.48 to the full management diagram shown in Fig. 2.47, which includes both the tactical and strategic level.

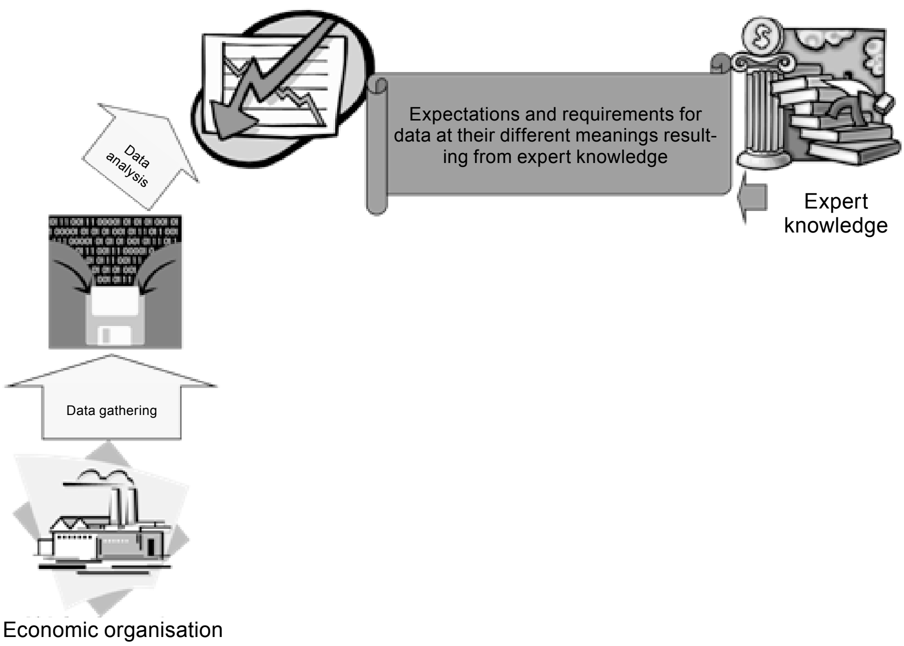

Fig. 2.51 The first step in the pursuit of a UBMSS system – providing the system with expert knowledge
Source: Own work

The first step to achieve this goal is to embed a large amount of expert knowledge in the system (Fig. 2.51).

Cognitive science, which is an interdisciplinary science dealing with the modelling of the mind and the natural and artificial information processing systems – can be very useful in constructing this resource of expert knowledge. Based on cognitive methods, we can also cause that the expert knowledge built into the system is not a dead resource, but a generator of hypotheses (on the interpretation of the economic situation) and the conditions to be met by indicators resulting from the analysis of input data, so that adoption of one of these hypotheses is justified.

Conditions resulting from expert knowledge (which by nature is built in a general and not specific manner) cannot be directly confronted with specific and very detailed indicators, resulting from the analysis of current and past economic data. Therefore, these analysis results must be first converted to the form based on the use of a special language (artificial, formal), which allows to describe the current economic situation in terms of abstract and general enough categories to allow for the confrontation of these data with the expectations deduced from the expert knowledge contained in the system (Fig. 2.52).

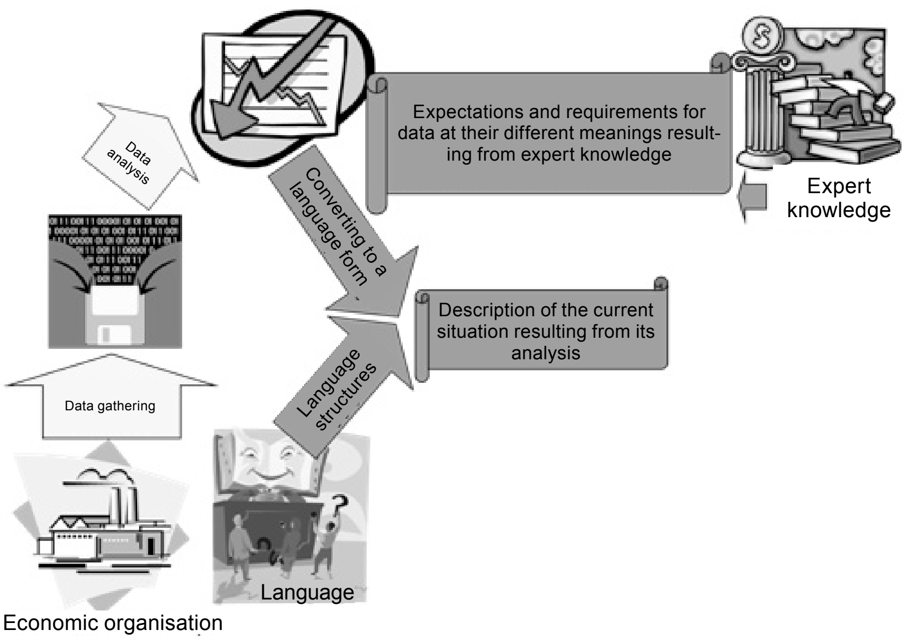

Fig. 2.52 Preparing to confront the results of analysis of current data with theoretical requirements arising from the expert knowledge
Source: Own work

In the situation prepared in a manner shown in Fig. 2.52, we must make the final comparison and draw conclusions. A process which in the context of earlier work has been called cognitive resonance (Tadeusiewicz and Ogiela, 2004), and in which a parser (syntactic analyser) of the language plays an important role, is useful for this task. The result of this comparison with a well-built parser is equivalent to finding (indicating) the proper semantic interpretation of data gathered, and this in turn leads to a obtaining a message that as a result of automatic understanding of the current macro- and microeconomic situation is directed to the strategic decision maker (Fig. 2.53). The complete structure of the UBMSS system is created in this way.

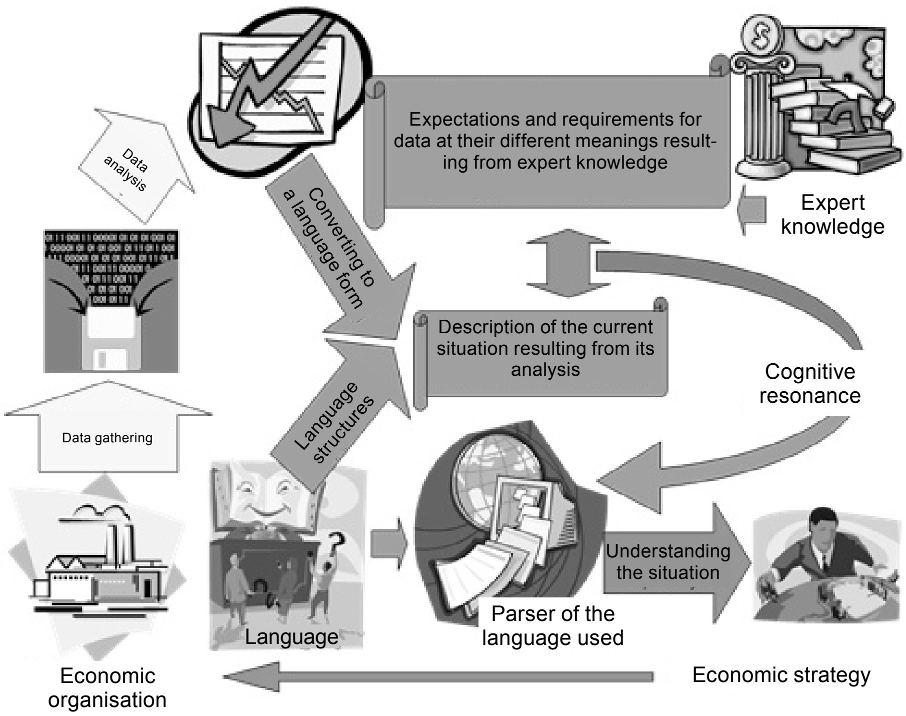

Fig. 2.53 The complete structure of the UBMSS system
Source: Own work

2.5.4. Summary and conclusions

The above content does not relate – in contrast to other chapters of this textbook – to the current situation in the field of information system design, but runs into the future, outlining a vision of systems that may be designed in the future, but certainly do not exist now. However, it seems that among the many specifics and details making up the content of this book, this little futurism and fantasy may also be helpful. For although many of the concepts and statements presented in this chapter may prove to be misguided, it is a fact that information systems are inevitably moving towards stronger penetration of the semantic side of the collected and analysed data. If progress will be achieved on this path, through the criticism and rejection of the concept and structure of the UBMSS system, among others, the presentation of this concept and this structure in this book can be regarded as fully justified.

3

ENTERPRISE SYSTEMS

3.1. PROCESS-ORIENTED ORGANISATION

In recent years, in the field of analysis and design of management information systems, methods derived from management science, as opposed to computer science, are gaining more importance. The keyword in modern methods of management is the concept of a process and horizontal organisation. In the early management methodologies, the approach based on a hierarchical division of the company into departments specialising in the implementation of certain functions was promoted. Such a construction of the organisation, and in particular its information system, led to information bottlenecks between isolated departments of the company. Seen from the customer perspective, they have to contact multiple departments, which implemented the service partially. Since there was not a single coordinator supporting the customer, the responsibility for the service was blurry. In times of the manufacturer market, functional (vertical) organisations could afford such treatment of customers.

In the age of the customer market, strong competition and market saturation of similar quality products, organisations can gain a market advantage through the quality of their customer service process. Modern management methods treat the concept of the business process as a key to processes-oriented enterprise management. A process is defined as (Hammer and Champy, 1996):

The set of activities that require a contribution on the input and giving a result on the output with a value for the customer.

In a process-oriented (horizontal) organisation, customer service processes, which give an added value to the customer, are the most important. The basis for the interest in processes has been created by such management methodologies as Business Process Reengineering (Hammer and Champy, 1996), Lean Management, or TQM.

The first of them suggested the revolutionary destruction of existing structures and designing of a process-based organisation. Subsequent methodologies also had similar goals but their achievement was more evolutionary in nature.

Among the management techniques that require the identification of business processes and their modelling and measurement, Gabryelczyk (2006) lists the following elements:

- Reorganisation changes within the organisation
- ISO certification

- Implementation of integrated management support systems
- E-business
- Defining the value chain between organisations
- The use of activity based costing
- Knowledge management systems
- Customer relationship management (CRM)
- Use of the Balanced Scorecard

In the process approach (Fig. 3.54), prior to designing changes in the organisation (a change in the information system is a design of this type), the business processes of the organisation should be mapped (documented) and then redesigned in accordance with the assumed objectives. Standard notations such as IDEF0, eEPC or BPMN (Business Process Modelling Notation) are used to describe the processes. The process approach is not a comprehensive analysis and design methodology, but it is a very effective method of ensuring the users and suppliers of IT tools understand how an organisation should operate after the implementation of planned changes. Business process modelling is done at the beginning of a project of implementing a new system. The model of information processes, which will be implemented in IT tools, should be built on the basis of a map of business processes.

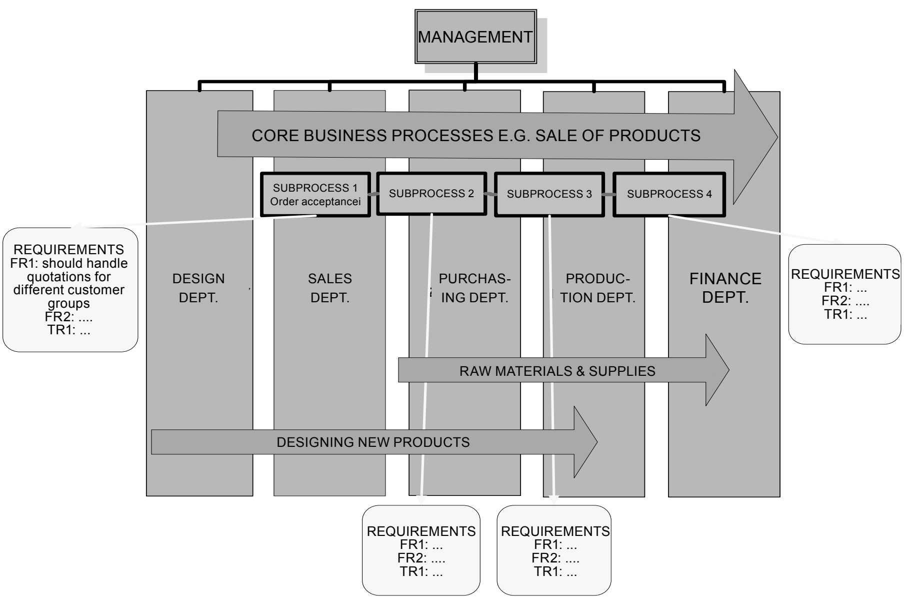

Fig. 3.54 Functional structure of an organisation with processes
Source: Own work

The advantage of the process approach is the focus on the strategic business purpose and the use of the business language of the organisation. Elements of information technology are left somewhat in the background. In order to prove the thesis of the popularity of this approach, we could point to the fact that the BPMN notation standard is also developed by the OMG organisation and perhaps will be included in future versions of UML. The process approach is a key concept on which the efficient and effective implementation of the enterprise systems discussed in this section depends.

3.2. THE CONCEPT OF AN ENTERPRISE SYSTEM

3.2.1. Review of the definition of an enterprise system

Today's integrated enterprise management systems (enterprise systems) are complex software packages that include mechanisms to support management of the entire enterprise and integrate all areas of the company's operation. These systems enable seamless integration of all information flowing within the company – such as information on finance, accounting, human resources, customers and information related to the supply chain. Enterprise systems are defined in various ways; the key definitions are listed in the following paragraphs:

- Enterprise systems are often identified as ERP systems (Enterprise Resource Planning) (Davenport, 1998).
- An ERP system is a business software package that enables organisations to: (1) automate and integrate the majority of its business processes, (2) share common data and practices across the entire enterprise, (3) generate and access information in a real-time environment (Deloitte Consulting, 1998).
- An ERP software package is an integrated set of modules (applications) that is ready to be implemented, supports all business functions of the company and allows dynamic configuration. It enables companies to process data in real time in an integrated, process-oriented and information-driven environment (Kale, 2001, pp. 24-25).
- A commercial software package that enables the integration of transactions-oriented data and business processes across the enterprise (Markus and Tanis, 2000, p. 176).
- A software package, which aims to integrate all business functions and processes and to provide uniform information and computer architecture for managing the entire company (Klaus et al., 2000).

- A comprehensive software package which offers the ability to integrate data and processes over the enterprise business functions (Brown and Vessey, 2003, p. 65).
- A software package that integrates organisational processes through shared information and data flows (Shanks and Seddon, 2000, p. 243).

3.2.2. Integration of the company as a result of using an enterprise system

Examples of definitions of an enterprise system presented above illustrate that the key features of this class of systems consist in integration of both data and processes. In particular, T. Gattiker and D. Goodhue (2000) define integration as a combination of (data) information and processes taking place in disjoint units of the company. Integration understood in this way can take place between companies, business units or between the areas (departments) of the company involved in various functions.

When examining the issue of enterprise integration in the context of the enterprise system, it is therefore convenient to make a division of integration into data integration and process integration (Volkoff et al., 2005). This division is associated with two basic characteristics of enterprise systems, i.e. the common central database and integrated business processes based on the so-called best business practice. Taking into account the context of integration, there are different integrations between the organisational units of the company depending on the relationships between them. There are three types of relationships that occur, respectively, between (Volkoff et al., 2005):

1. Various business units of the company performing similar functions,
2. Various stages of the business process, and
3. Different functional areas of the company performing various functions within the same organisation.

These three types of relations correspond to the types of interdependencies between different organisational (business) units, which can be divided into pooled, sequential and reciprocal interdependence (Volkoff et al., 2005). Pooled interdependence means that each business unit makes its own independent contribution to the common result, therefore it both supports the other units and is supported by them. In this dependence, the units do not interact directly with each other. In the sequential interdependence, the outputs from one business unit are the inputs for another unit. Finally, in the reciprocal interdependence, the outputs of the several units become inputs for one another.

The occurrence of reciprocal interdependence is associated with the occurrence of the other two, while the occurrence of sequential interdependence implies the presence of a pooled interdependence. The difficulties of coordination increase with the change of the type of interdependence from the pooled through sequential to reciprocal interdependence. Table 3.8 shows the characteristics of enterprise integration in the context of the application of an enterprise system. Different types of integrations are listed, resulting from the types of interdependencies and forms of associated relations. Characteristics shown in this table include the division into the integration of processes and data.

Table 3.8 Types and characteristics of integration in the context of the application of an enterprise system

Type of integration	Processes	Data
Pooled	Standardized processes	Standardized data
Sequential	Closely related processes	High level of accuracy and consistency of data required
Reciprocal	Processes coordinated in a continuous manner	Sharing data in real-time

Source: Own work based on (Volkoff *et al.*, 2005 p 115)

Enterprise systems, having a central database and built-in standard processes, allow for coordination between individual business units of the company, thus support integration. In the case of the simplest type of integration, i.e. the pooled integration, the distinctive features include unification of processes and data, which are the most basic characteristics of an enterprise system. With the change of the type of integration into a more complex one, additional features appear. Sequential integration is characterised by the need to closely link processes and high accuracy and detail level of data across all activities within the processes. Reciprocal integration is related with the need to share data in real time and continuous coordination of processes (Soja, 2006b; Volkoff et al., 2005).

3.2.3. Key features of enterprise systems and conditions for their application in an enterprise

Basic features of enterprise systems are related to the presence of a central database, the so-called standard models of business processes (reference models), and the need for system configuration (see Davenport, 1998; Kremers and van Dissel; 2000, Kumar and van Hillegersberg, 2000; Scheer and Habermann, 2000; Soja, 2006b).

- *Central database* – provides a single source of data and helps different applications and system modules to exchange data. The database collects and stores data from various business areas. Use of a single database greatly improves flow of information through all areas of business operation. The source of data for the database and its users are the organisational units and the applications operating in them in various functional business divisions. New information is entered only in one place in the database, which automatically updates all related data. The presence of a central database is a major challenge for large companies where various organisational units generate fragmentary information with different levels of detail.
- *Standard models of business processes* – delivered by the majority of integrated packages, they are defined based on the best solutions worked out in business practice. The proposed business models, called reference models, contain the data structures, models, processes, and organisational structures used. Reference models provided by vendors of enterprise systems allow buyers to use the knowledge and best practices developed in a given industry. The company purchasing a ready-to-use integrated package receives the entire knowledge on the solutions in the industry and is somehow forced to implement it. However, the question remains whether the reference models offered by the system are the best solution for a particular company, as well as the related issue concerning the source of competitive advantage, if the majority of enterprises in the same industry will use the same organisational solutions.
- *System configuration – a* high level of generalisation of enterprise systems, due to the evolution consisting in increasing the number of parameters, options, and configurable functionality in the systems, causes the need for configuring the system as a part of its implementation in a particular enterprise. During this process, companies have to work out a compromise consisting in adapting capabilities of the system to the needs of the enterprise. This is done under the assumption that a greater modification (customisation) of the system is impossible, and the configuration of the system boils down to the choice of modules and filling in the configuration tables. This is not a simple matter, which is well illustrated by the fact that the SAP R/3 system has over 3000 configuration tables. During the process of adapting the system to the needs of the organisation, companies should be aware of the potential integration benefits and bear in mind that greater customisation of the system causes a threat to integration.

Features of enterprise systems cause that their use in organisations is associated with a number of issues that are both organisational and technology-related.

We will briefly discuss the most important determinants of an enterprise system implementation project below (Markus and Tanis, 2000).

- *High cost and risk* – observations of the implementation market show that the implementation of an enterprise system is associated with a high risk of failure. This market has seen many failures and the question arises regarding the profitability of investments in an enterprise system.
- *Technological advancement* – this characteristic of an enterprise system results in the necessity for a series of activities such as system selection, configuration, adaptation to the needs of the company, business modelling, integration with other systems and architectures.
- *A challenge for management – the* management of an implementation project is a challenge primarily because participants from several departments of the company are engaged at different organisational levels and even coming from many different organisations. Implementation of an enterprise system also affects the organization of the company's information system and poses new challenges connected with the qualifications of employees and the turnover of staff.
- *Impact on the enterprise* – the implementation of an enterprise system is related to the question of to what extent the company is able to exploit the potentially huge opportunities of an enterprise system and how efficiently is the enterprise system used in companies.
- *Integration* – is related to the question of to what extent the company should be reorganised to adapt to the enterprise system. It also raises the question of the role of the provider of a system and implementation services in the implementation process and the issue of dependence on a supplier.

A variety of factors related to the use of enterprise systems are well summed up in the article by C. Stefanou (2001), which outlines the following factors influencing the complexity of assessment of an enterprise system:

- An enterprise system is a system which is both operational and strategic in nature.
- The use of an enterprise system in the enterprise involves many stakeholders, who are also outside the organisational boundaries of the enterprise.
- The use of an enterprise system is closely related to the occurrence of non-measurable costs and benefits.
- The implementation of an enterprise system involves major organisational changes.
- The benefits and costs associated with the use of an enterprise system occur throughout the system life cycle.

3.2.4. The architecture and the basic modules of an enterprise system

The general architecture of modern enterprise systems (also called ERP II systems) can be represented as a set of components (Fig. 3.55). An ERP system is a central element of the architecture and is surrounded by the set of so-called *corporate* components (Moller, 2005):

- SCM (Supply Chain Management)
- CRM (Customer Relationship Management)
- SRM (Supplier Relationship Management)
- CPM (Customer Performance Management)
- HRM (Human Resource Management)
- PLM (Product Lifecycle Management)

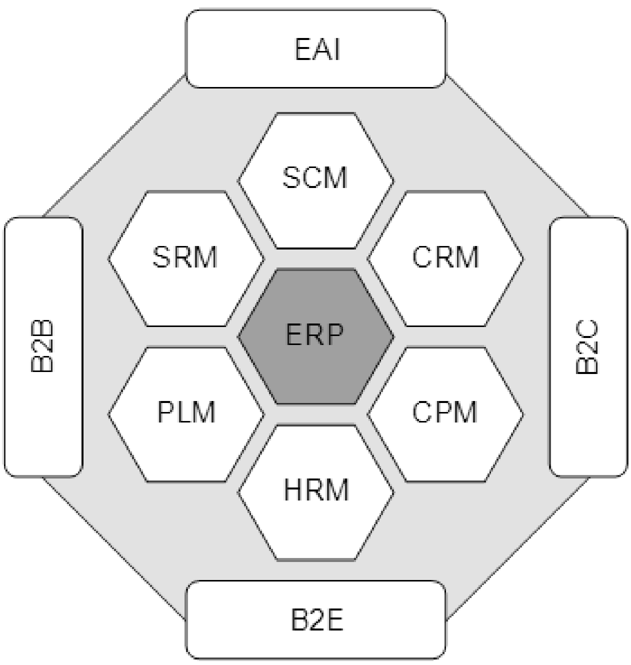

Fig. 3.55 Component architecture of the enterprise system
Source: Own work based on (Moller, 2005, p. 490)

The outer layer of the model is created by components that handle the communication and integration between the enterprise system and external entities. The components handling cooperation include (Soja and Stal, 2008):

- *Business-to-consumer* (B2C) – refers mainly to transactions involving sales via the Internet, mostly to individual customers, but may also relate to business partners.
- *Business-to-business* (B2B) – refers to streamlining and automating transactions between business partners, focusing on the supply process with the use of mechanisms of electronic auctions, markets, and catalogues.
- *Business-to-employee* (B2E) – handles providing employees with personalised access to current corporate resources using a suitable portal.
- EAI (Enterprise Application Integration) - a platform for the integration of the enterprise system with other systems inside and outside the organisation.

The presented architecture illustrates the openness of today's enterprise systems to the outside world and the need for mutual communication. The main approach used in the integration of modern enterprise systems is a service-oriented architecture (SOA), whose goal is to achieve a convenient connection between the interacting systems. A service is defined as a task performed by the service provider in order to achieve the end result desired by the recipient (customer) of the service. In this approach, the main emphasis is on defining a set of services, with functionality consistent with user requirements. On the basis of services created, it is possible to construct distributed systems and applications, independent of the software platform used, communicating with services using defined protocols and data formats.

3.2.5. Selected modules of the enterprise system using the example of SAP R/3

Modern enterprise systems cover practically the entire enterprise and support a number of inter-organisational activities. Below, we briefly discuss the most important modules of the system, supporting key business areas, using the example of SAP R/3, offered by SAP – a leading provider of enterprise systems on a global scale. Discussion of the modules includes an emphasis on core business processes supported by a given module, with the indication of connections with other system modules (Kale, 2001, pp. 189-224).

Finance and Controlling (FI-CO)

The solutions offered by SAP in finance and accounting are aimed at internationalisation of accounting, which involves support for multiple currencies and languages and flexibility of solutions to adapt to specific country regulations. Processes offered by the Finance (FI) module include: posting in the General Ledger and its closing, accounts receivable, accounts payable, fixed asset management, consolidation, and special books. Processes available in the Controlling (CO) module include: cost type accounting, cost centre accounting, profit centre accounting, product cost planning, activity drivers costing, income statements, ABC costing.

The FI-CO module is the most central module in SAP; virtually every system module communicates with FI-CO. In particular, the most critical system modules connect to the FI-CO in the following areas:

- Sales and Distribution (SD) – credit control, shipping and return of goods, taxation and pricing schemes, outstanding balance, profitability analysis, dunning, prepayment, discounts
- Materials Management (MM) – orders, purchase invoices, supplier payments, quality control, inventory control, inventory differences, freight and insurance fees etc.
- Production Planning (PP) – production orders, production variances, finished goods, work in progress, etc.
- Human Resources (HR) – salaries, benefits, delegations, advances and loans, bonuses, etc.

Sales and Distribution (SD)

The SD module provides the ability to effectively manage the operations relating to the sale and distribution of goods and services. Key business processes offered by the SD module include: processing of request for proposals and customer queries, processing of offers, placing sale orders, supply planning, monitoring availability, pricing schemes, credit control, invoicing, shipping, customer payments, inventory management, transportation management, and sales reports.

Connections with other key modules cover the following areas:

- Materials Management (MM) – checking availability, delivery schedules, materials shipment, transfer of materials between facilities, material determination and materials substitution, reorder points and returns
- Production Planning (PP) – checking availability, sales and production planning, transfer of orders to the PP module
- Finance (FI) – credit control, materials shipment, return of materials, invoicing, taxes, prices, arrears, and profitability analysis

Materials Management (MM)

The MM module is a system of inventory management which can handle both the stocks of goods and orders from subcontractors. MM offers the following business processes related to the management of materials: purchase of materials, requests for pricing, quotation, order processing, receipts for goods, checking invoices, payments, overdue payments, evaluation of suppliers, and quality control.

MM fully integrates with other modules in the area of logistics and finance. In particular, connections with other key modules cover the following areas:

- Finance and Controlling (FI-CO) – orders, purchase invoices, supplier payments, materials transfers, quality control, stock records, inventory differences, etc.
- Sales and Distribution (SD) – availability control, scheduling of deliveries, credit control, materials shipment, inventory transfers, material determination and substitute materials, reorder points, returns, etc.
- Production Planning (PP) – list of requirements, delivery schedule lines generated by the MRP, inventory collections based on confirmations and production orders, etc.

Production Planning (PP)

The PP module is a central part of the enterprise system, which focuses on various activities in many areas of the enterprise. Customer requirements are processed into demand requests, which constitute input data for the production plan. It is available as a master production schedule (MPS) and is balanced based on a rough plan of production capacity. As a result of MRP operation, there is an automatic generation of production orders and purchase orders for the relevant materials. Production orders start the production activities. SAP is able to handle all types of production: continuous production, batch production, production for an inventory, assembly according to the catalogue and project production. PP includes the following business processes: sales planning and rough production planning, rough capacity planning, demand management, MPS, material requirements planning (MRP), long term planning, capacity planning, production control and plant maintenance.

Connections with other key modules of the system cover the following areas:

- Finance and Controlling (FI-CO) – production orders, production variances, finished goods, work in progress, etc.
- Sales and Distribution (SD) – availability control and sales forecasting
- Materials Management (MM) – a list of demands and supplies, delivery schedule lines, etc.
- Human Resources (HR) – attendance, working time, wages, bonuses, etc.

Human Resources (HR)

The HR module enables optimal utilisation of human resources across the enterprise, supporting areas related to recruitment, staff development, work time management, wages, bonuses, etc. The key business processes offered by HR include: organisation management, skills management, career planning, event management, planning of human potential, personal cost planning, personal data management, development of human resources, payroll, work time management, and business travel management.

Connections with other key modules of the system cover the following areas:

- Finance and Controlling (FI-CO) – attendance, exits, payroll accounting, advances, training expenses, business travel, etc.
- Sales and Distribution (SD) – meetings, committees, etc.,
- Production Planning (PP) – attendance, work time, motivational benefits, etc.

3.3. THE BASIC CLASSES OF ENTERPRISE SYSTEMS AND THEIR EVOLUTION

3.3.1. Inventory control systems

Enterprise systems are a solution, which was a result of the evolution of management support systems (Soja, 2005). Before they were created, there were systems, which evolved into newer and more advanced solutions. As a result of this evolution, a new system was built based on the existing one, supplemented by new functions and features, which included the properties of its predecessor. Each new system covered and integrated more and more business functions (Parys, 1999a). The evolution of the enterprise systems was accompanied by changes in computer technology and software that allowed to build more and more complex, versatile, and comprehensive systems at an increasing degree of integration. The importance of information technology has also changed, from the role of supporting the operations of the company to a strategic role.

The initial stage of the development of enterprise systems took place in the 1950s. The first materials management systems appeared in enterprises at that time. These systems, supported by software based on statistics and common sense methods, automated the activities carried out in the management of inventory (Parys, 1999b; Popończyk, 1996).

In the 1960s, inventory control techniques appeared based on forecasting methods and the reorder point method. They used the information on inventory consumption in prior periods as a basis for planning and controlling inventory levels in the future (Parys, 1999a).

The development of computer technology and the increase in processing capacity, thus the speed of calculations, meant that over time it became possible to eliminate problems associated with time consumption and labour intensity of calculations. This enabled the merger of activities such as forecasting, defining the size of orders and delivery times, determining inventory levels, etc. into a single comprehensive system.

3.3.2. MRP systems

As a result of the development of computer technology, the MRP system (Material Requirements Planning) was created. The real development of MRP systems began in the early 60s of the twentieth century with the adoption of quantitative methods of management supported by computer technology (Durlik, 1996). MRP systems were the result of search for better methods of ordering and obtaining of materials and components for production process (Wallace, 1990).

MRP combines the worked out production schedule with a list of materials necessary to manufacture the product, examines the production inventory and determines which parts and raw materials have to be ordered and when, so that they are stored in the manufacturing process for as short as possible. Taking into account when different parts of the final product should be manufactured according to the schedule and the necessary periods of receipt of the materials, MRP distributes the inventory replenishment orders in time in such a way that parts and materials are available in the manufacturing process at the point in time when they are needed on the workstations (Durlik, 1996).

The MRP system therefore requires three basic sources of information, which are (Meredith, 1992): production schedule, product structure and data on inventory. For proper operation, high accuracy and correctness of the data on the structure of the product and the level of inventories is necessary. If the data are inaccurate, the MRP system schedules the components incorrectly and does not calculate the right quantities for the order (Fogarty et al., 1989).

3.3.3. The closed-loop MRP system

To improve the efficiency of production planning, it is necessary to compare the plans with the results of their implementation and use of this information in subsequent stages of planning. However, the MRP system does not offer such opportunities. The mechanism proposed by this system fails to provide feedback on the implementation of the planned orders. What's more, the basic MRP system includes only that part of the control of operating activities, which concerns the flow of materials and does not consider the implications in relation to production capacity (Muhlemann et al., 1995).

Enhancing the basic MRP system with the mechanisms that provide feedback on the implementation of the proposed orders and the capacity planning mechanisms, among others, led to the creation of the closed loop MRP. Capacity planning mechanism is powered by information from the master production schedule and the material requirements plan, which is the effect of the operation of the MRP mechanism. Its task is to assess whether the plans included in the MPS and MRP are feasible, i.e. whether the company has sufficient production capacity to implement the plans. The result is a plan consisting of orders which should be fulfilled.

The following tools are available for monitoring of the implementation of planned orders: purchasing control and production control, which provide feedback for the preparation of the MPS, creating the material requirements plan and planning capacity. This feedback allows the company management to check whether any corrective action is needed, and if such a need arises, it makes it easier to take appropriate steps (Fogarty et al., 1989).

3.3.4. MRP II system

MRP II (Manufacturing Resource Planning) is the next stage in the evolution of enterprise systems. Since the acronym of this name is identical to the one used for its predecessor, the Roman numeral II was added to distinguish it. Extending the direct predecessor of the MRP II system, i.e. the closed-loop MRP, was based primarily on the connection with the sales processes, linking with financial and long-term business planning, and the introduction of possibilities for simulation (Wallace, 1990).

The MRP II system covers the entire enterprise, integrates sales, finance and manufacturing. It processes the demand for manufacturing resources (i.e. services, equipment, staff, and materials) into financial requirements and presents the results of production activities in financial terms. This processing capacity helps in estimating the company's financial ability to implement the plan and in the presentation of the financial aspects of the production plan in the form of appropriate indicators (Fogarty et al., 1989).

3.3.5. ERP systems

Extending the scope of MRP II to ERP (Enterprise Resource Planning) is a consequence of the approach that ensures rationality, which is dominant in Western culture and management traditions and is based on the so-called capital account (Greniewski, 1997). The enhancement of the MRP II system included financial procedures, such as cost accounting, management accounting, cash flow, controlling, etc. The introduction of these procedures to the system enhances its functionality and allows for planning and production control, based not only on quantitative indicators, but also financial measures (Parys, 1999a).

The essence of ERP systems is a more comprehensive business analysis and functional scope, which must cover all areas of business activities within the entire logistics chain (Adamczewski, 1999). Additional domain areas in the ERP system include: distribution management, asset records and settlement, human resources and payroll, marketing, bidding, maintenance service management.

Operation of an ERP system can be illustrated using the example of sales operations. In the ERP system, when the salesperson takes an order from the customer, he or she has access to all information needed for order fulfilment, including the history of orders associated with the customer, credit granted, stock inventory, and delivery schedule. Each employee in a company can obtain exactly the same information and have access to the same database that contains data about the order. When one department of the company completes work on the given order, it is automatically transferred through the system to the next department for further processing. In order to find out what is happening with the order at any point in time, one must simply log on to the system (Koch, 2001).

3.3.6. The concept of ERP II systems

Modern enterprise systems are the result of further evolution of ERP packages towards the support of various activities that take place between organisations and towards the support of inter-organisational cooperation (Soja and Stal, 2008). This evolution was aimed at eliminating a number of limitations of ERP systems related with inter-organisational integration. These restrictions include (Akkermans *et al*., 2003): insufficient support for the processes that go beyond the enterprise, the lack of flexibility in relation to the ever-changing needs of the supply chain, the lack of functionality that goes beyond the management of transactions and the lack of modular and open system architecture.

As a result of evolution, enterprise systems began to support company activities related to customer service (front-office), as well as inter-organisational activities, including supply chain management and customer relationship management (Davenport, 2000). Enterprise systems started uniformly linking front-office activities (i.e. sales, marketing, customer service) with back-office activities (operations, logistics, finance, human resources) in order to gain a competitive advantage (Chen, 2001).

Modern enterprise systems are often called ERP II systems, which is a concept developed by the GartnerGroup in 2000. ERP II is defined as a business strategy and a set of industry-domain-specific applications that build consumer and shareholdervalue by enabling and optimising enterprise and inter-enterprise, collaborative-operational and financial processes (Bond *et al*., 2000).

Summarising, ERP II systems are enterprise systems supporting the enterprise resource planning methodology (MRP II), allowing for planning and financial management, and supporting contacts with the outside world by allowing communication with the system via the World Wide Web, and offering the function of customer relationship management (Lech, 2003).

3.3.7. Trends and directions in the development of enterprise systems

Enterprise systems are continuously evolving in terms of their technology and functionality. The directions of development and improvement of enterprise systems include (Kumar and van Hillegersberg, 2000; Soja, 2005):

- Integrating new areas of activity by adding new modules (applications) to the system or by developing new versions of existing modules.
- Taking into account the industry-specific solutions supported so far by specialised software only (such as insurance activity, banking, financial services, building industry).
- Providing mechanisms to optimise production in the logistics chain and adapting the supply chain to the needs of individual companies.
- Replacing the the packaged monolithic architecture with systems consisting of a core module offering basic minimum functionality and a set of components that can be independently developed or acquired from outside providers.
- Outsourcing of enterprise systems and providing systems using the SaaS model (Software as a Service) by offering customers the access to the system (via a web browser), while the system database and the computers are in a remote data centre managed by the provider.
- Use of platforms and technologies to implement e-commerce via the Internet.

3.4. SUCCESS IN ENTERPRISE SYSTEM ADOPTION

3.4.1. The concept and the evolution of an information system success

The success of an information system (IS) is an extremely important concept, whose understanding contributes to the effective implementation and use of IS in an organisation. The concept of IS success involves a number of topics related to system performance, efficiency of the implementation process, as well as the level of the information system's support for the activities of individual employees and the organisation as a whole (Delone and McLean, 1992; Soja, 2010a).

Over the years, the IS success attracted the attention of many researchers who have generally focused on individual aspects of the impact of the IS on the company. The most important single aspect of the IS success found in the literature is user satisfaction when using the system (Doll and Torkzadeh, 1988). In addition to approaches investigating individual aspects of IS success, models that considered several criteria have also appeared. The most interesting proposals of this type include the approach proposed by K. Lyytinen and R. Hirschheim (1987), which very well illustrates the complicated nature of the issue of information system success and the fact that the achievement of success is a great challenge for the organisation. The authors define the success of the project in the area of information systems as follows (Lyytinen 1988; Lyytinen and Hirschheim, 1987; Kuraś and Zając, 1999; Soja, 2010a):

- The project has been completed and implemented.
- The project scope is not limited to that initially intended.
- Users accept the system and fully use it as intended.
- Operating parameters of the system are consistent with the planned ones or better.
- Implementation of the system yielded the expected results, which has been thoroughly investigated.
- The system has achieved the planned (or greater) efficiency within the prescribed period.
- The system is compatible with the objectives of the organisation and supports their achievement.

3.4.2. Multidimensional IS success model by DeLone and McLean

Research related to the concept of IS success have been comprehensively analysed by W. DeLone and E. McLean (1992), who proposed the most meaningful model of IS success as a result. In their studies, Delone and McLean organised the existing concepts related to IS success and proposed their taxonomy captured in an interactive model. The purpose of the DeLone and McLean (D&M) model was the synthesis of existing research on the success in the field of information systems and proposing research frameworks for future work.

The DeLone and McLean model contains six interrelated dimensions of success (Delone and McLean, 1992):

1. System quality (SQ)
2. Information quality (IQ)

3. System use (USE)
4. User satisfaction (US)
5. Individual impact (II)
6. Organisational impact (OI)

Delone and McLean presented the proposed dimensions of a success in a cause and effect model, as illustrated in Fig. 3.56. The cause and effect relationship between variables is depicted by arrows. It is related to the postulated effects of changes in the source variable on the increase or decrease of the target variable indicated by an arrowhead.

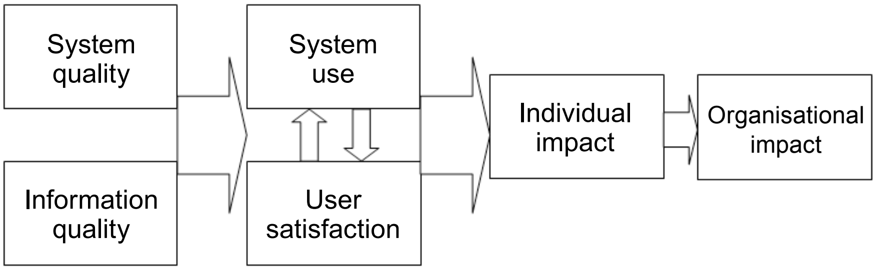

Fig. 3.56 The DeLone and McLean information system success model
Source: Own work based on (DeLone and McLean, 1992, p. 87)

The main conclusions for the D&M model were expressed by the authors as follows (Delone and McLean, 2002).

- The success of an information system is multi-dimensional in character, and its individual components are interdependent. This requires special attention when defining and measuring each aspect of success. The probable interaction between various dimensions of success should be measured.
- Selection of success dimensions and accompanying measures should be consistent with the objectives and context of empirical research. If possible, tested and proven measures should be used.
- Despite the multidimensional nature of the IS success and existing dependencies between dimensions, an attempt should be made to reduce the number of measures used for measuring success. The purpose of this process is greater opportunity to compare and verify results.
- Further empirical studies should focus primarily on developing measures related to the organisational impact.
- The proposed model requires further work and verification before it can serve as a basis for selecting appropriate measures of IS success.

The D&M success model has met with great interest and was subjected to many analyses and modifications. Among the dimensions of success proposed in the D&M model, the dimension associated with the use of the system raised the greatest doubts (Seddon, 1997; Gable et al., 2003). These doubts are related to the issue of whether the use of the system in a given organisation is mandatory. When this happens, the dimension determining the extent of system usage provides us with little information about the success of the system (Robey, 1979; Welke and Konsynski, 1980). We must therefore be aware that the analysis of the system usage is meaningful only when system usage by users is not obligatory (Delone and McLean, 1992, p. 68).

Referring to the results of numerous studies based on the original D&M model, and in order to eliminate its deficiencies, Delone and McLean proposed a modified version of their IS success model (Fig. 3.57). In the improved version of the model, the authors support the conclusions proposed in the original version of the model and present some modifications, which can be summarised as follows (Delone and McLean, 2002):

- Introducing an additional variable Service Quality.
- Combination of two variables describing the impact of the system (i.e. individual impact and organisational impact) into a single variable Net Benefits, which represents the benefits of using the system.
- Postulating the impact of the variable *Benefits* on the dimensions describing user satisfaction and system use.
- Change in the understanding of the *System Use* variable in order to enable to analyse the intention to use the system by users.

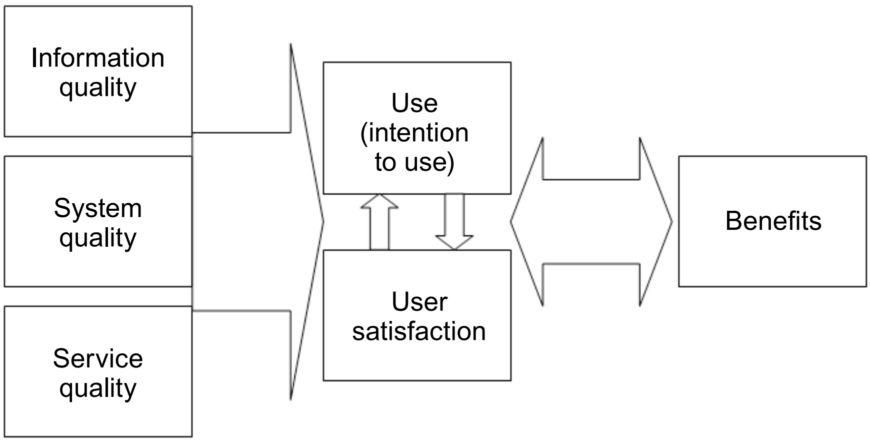

Fig. 3.57 Modified DeLone and McLean IS success model
Source: Own work based on (DeLone and McLean, 2002)

3.4.3. DeLone and McLean model in the area of enterprise systems

The DeLone and McLean Information System Success Model also includes topics related to the implementation of enterprise integrated systems. In this area, particularly interesting are the studies conducted by the Australian *Queensland University of Technology* (Gable et al., 2003; Sedera et al., 2004), in which researchers gathered the opinions of several hundred respondents from dozens of government agencies in Australia. The results suggest that success in the field of enterprise systems is a multidimensional concept and includes four distinct and interrelated dimensions (Fig. 3.58).

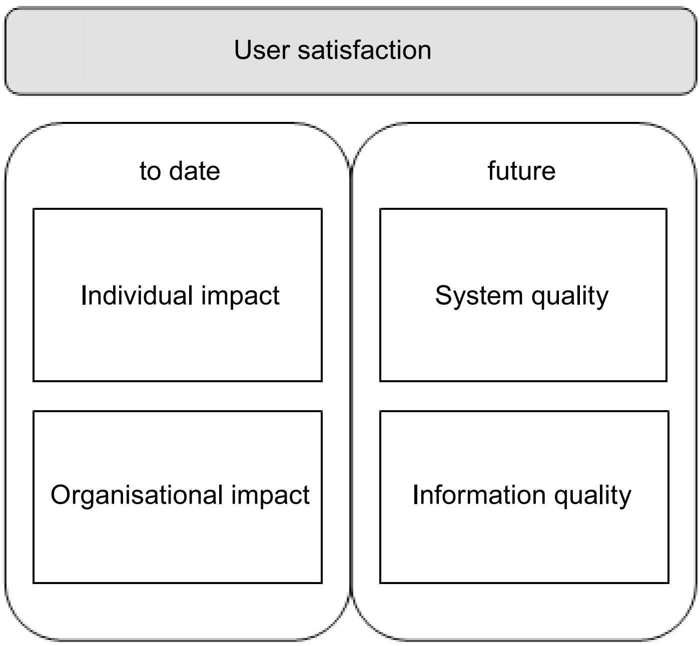

Fig. 3.58 DeLone and McLean IS success model in the area of enterprise systems
Source: Own work based on (Gable et al., 2003, p. 586; Soja, 2010a)

The proposed model includes dimensions known from the original approach of DeLone and McLean: individual impact, organisational impact, system quality, and information quality. It is worth noting that these dimensions are grouped in a slightly different manner than in the D&M model. The first group contains the dimensions related to the impact and illustrates the benefits gained at the given time from the application of the system. The second group contains the dimensions related to quality and illustrates the future potential of the system and the expected benefits from the use of the system in an organisation. Another important proposal is to treat customer satisfaction as a general measure of success, not as one of the dimensions of success, as it is in the D&M model.

Table 3.9 Measures for the dimensions of enterprise system success

Impact on the individual	System quality
learning	data accuracy
awareness	data currency
decision making effectiveness	database contents
individual productivity	ease of use
	ease of learning
	access
	user requirements
	system features
	system accuracy
	flexibility
	reliability
	efficiency
	sophistication
	integration
	customisation
Organisational impact	**Information quality**
organisational costs	importance
staff requirements	availability
cost reduction	usability
overall productivity	understandability
improved outcomes/outputs	relevance
increased capacity	format
e-government	content accuracy
business process change	conciseness
	timeliness
	uniqueness

Source: Own work based on (Gable *et al.* 2003 p 582; Soja, 2010a)

In addition to outlining the dimensions of the enterprise system success, the authors proposed the method of measuring individual issues. Each of the dimensions can be captured through a set of few to a dozen of variables, as shown in Table 3.9. Most suggested variables are applicable to the general concept of information system success. Only few variables were added due to the nature of enterprise systems, as shown in bold in the table. In order to collect data related to the enterprise system success, the authors use a questionnaire, which contains dozens of questions corresponding to the proposed variables.

3.4.4. User satisfaction as a general measure of the success of enterprise system implementation

Impact of the enterprise system on an enterprise is a complex and multidimensional topic, which is suggested by, among other things, the definition of the measures of success dimensions presented above. Therefore, in order to thoroughly examine all the dimensions of success, we meet difficulties associated with the need to use complex and labour-intensive research instruments. However, as was also suggested by the mentioned study by Gable et al. (2003), the situation is largely simplified by the use of user satisfaction as an overarching measure of success.

Table 3.10 Measurement of user satisfaction using the EUCS research instrument

Content
Does the system provide the precise information you need?
Does the information content meet your needs?
Does the system provide reports that seem to be just about exactly what you need?
Does the system provide sufficient information?
Accuracy
Is the system accurate?
Are you satisfied with the accuracy of the system?
Format
Do you think that the output is presented in a useful format?
Is the information clear?
Ease of use
Is the system user friendly?
Is the system easy to use?
Timeliness of information
Do you get the you need in information you need in time?
Does the system provide up-to-date information?

Source: Own work based on (Doll and Torkzadeh, 1988, p. 268; Soja, 2010a)

User satisfaction with the enterprise system is one of the most important determinants of the success of the implementation of this class of systems. This is because the accomplishment of the benefits from the enterprise system investment depends on the efficient use of information technology and satisfaction of this system's users (Somers et al., 2003). Thus, user satisfaction can be used as a measure showing the result of applying the enterprise system in a company.

There are several proposals for measuring user satisfaction, among which of particular interest is EUCS (End-User Computing Satisfaction) (Doll and Torkzadeh, 1988). EUCS is a research instrument consisting of 12 elements grouped into five groups related to content, accuracy, format, ease of use and relevance of information. The data related to these elements are collected using questions presented in Table 3.10. Answers to questions are collected in a five-point scale, where 1 means "completely disagree" and 5 "completely agree". It should be particularly emphasised that EUCS was successfully verified in the field of ERP integrated systems (Somers et al., 2003).

3.4.5. The importance of success depending on the phase of the life cycle of the enterprise system

Implementation of the enterprise system is a multi-stage project, under which the enterprise must perform a variety of activities, whose character may depend on the project implementation phase. This causes that the determinants of the implementation process and the understanding of success may depend on the actual stage of the project. Process approach to enterprise system success consists in taking into account the life cycle of the enterprise system in the understanding of the implementation success (Markus and Tanis, 2000).

Application of the enterprise system in the enterprise can be divided into the following three stages (Markus et al., 2000):

1. *The project stage* (implementation), during which the system is configured and launched throughout the company.
2. *The shakedown stage* (reorganisation), during which the company makes the transition from the launching of the system to normal operations.
3. *The onward and upward stage* (*continued use of the system*), during which the company gains most of the benefits from the system and is planning further steps in its development.

According to the process approach, there may be different measures of success at each *stage* of the project. In particular, in the implementation (project) *stage*, the success measures are typical measures for general projects and may include the following indicators:

- Cost of the project in relation to the planned budget
- Duration of the implementation compared to the schedule

- Installed system functionality in relation to the plan

The concept of success at the project stage is related to the understanding of the project, which can be defined as an undertaking with a clearly defined time frame. In other words, it is an activity that has a pre-planned end. Project implementation naturally requires the company's resources, which have the form of planned and allocated funds. Finally, the project generates a given effect (result), which we can compare to the original intentions. According to this, the success of the project should be defined in three dimensions and the results achieved should be examined compared to the planned values. Thus, a project which has achieved complete success is an undertaking, which was completed according to the planned schedule, fits in the planned budget and achieved the desired results (or in other words, provided a solution with the assumed quality).

The reorganization stage requires taking into account measures illustrating the organisational benefits of using the system. Examples of such indicators include (Markus et al., 2000; Soja, 2010a):

- Short-term changes in key performance indicators, which occurred after the commissioning of the system (such as operating labour cost).
- The time required to achieve normal or expected values by the key performance indicators (after the initial drop).
- Short-term impact of the use of the system on the company and its customers and suppliers (e.g., the average time of accepting an order by phone).

At the last stage, or the stage of using the system, the company is already functioning in a normal way and measurable business results arising from the use of the system should be expected. At this stage, the sample measures may include the following:

- Return on investment
- The achievement of the planned quantitative effects from the package implementation (such as IT cost reduction, reduction of inventory and storage costs)
- The achievement of planned qualitative objectives of implementation (e.g., creating a single corporate image to customers)
- Further improvement and increase of business benefits after having achieved the expected results
- Ease in adopting new versions of the system and other new IT products
- Improved business processes
- Better decision making

3.4.6. The concept of a relative success

Considering the concept of the enterprise system implementation success, we should take the condition of the company and its market situation into account. This is linked to the notion of a relative success (also called the optimal success), related to the best results the company can achieve given its particular business situation (Markus and Tanis, 2000, p. 184). Depending on how ambitious were the goals adopted by the organisation as the planned result of the system implementation, the optimal success may significantly deviate from the goals set for the implementation of the system, in both positive and negative sense.

Relative (optimal) success of the implementation of an enterprise system can be dynamic, i.e., the results achieved by the company may change over time due to changes in the conditions of business activity. Therefore, when speaking about the success of the enterprise system adoption in the company, we should understand this concept as the current estimate of the success of the system in the enterprise, measured using indicators appropriate for the current implementation phase of the system (Markus and Tanis, 2000, p. 187).

3.5. MOTIVATIONS FOR ENTERPRISE SYSTEM ADOPTIONS

3.5.1. Reasons for adopting the enterprise system depending on the size of the company

Companies deciding to implement an enterprise system are guided by a variety of motives, which may depend on the type of implementation and the type of business. The most common division is the distinction between small and large enterprises. This criterion is most often used by experts and researchers when differentiating the conditions of the implementation of an enterprise system. This criterion also applies in the case of motivations for an enterprise system implementation.

Large enterprises with complex organisational structures indicate the same reasons as the small companies, however, there is also a group of reasons specific to them (Markus and Tanis, 2000). Table 3.11 summarises the reasons for the use of enterprise systems in companies using the division into technical and business reasons, taking the size of the company into account. Technical reasons include issues related to the system used at the enterprise and the IT infrastructure, and business reasons are related to the business activity carried out by the company.

Table 3.11 Reasons for adopting the enterprise system depending on the size of the company

	Small companies / Simple structures	Large companies / Complex structures
Technical reasons	Solving the year 2000 problem and similar problems	Most common reasons for small companies, plus:
	Integration of applications with different functionalities	Consolidation of many different systems of the same type (e.g. general ledger packages)
	Replacement of hard-to-maintain programs	
	Reduced software maintenance burden through outsourcing	
	Removal of multiple data entry and related errors and problems with data analysis	
	Improving IT architecture	
	Better use of technological restrictions	
	Reduction of computer operating costs	
Business reasons	Adaptation to the company's growth	Most common reasons for small companies, plus:
	Introduction of multiple languagesand multiple currencies	Providing integrated IT support
	Improving informal and/or inefficient business processes	Standardisation of various patterns of naming, numbering and coding
	Data cleanup through standardisation	Standardisation of procedures in different departments/locations
	Reduction of business operating and administrative expenses	Creation of a single corporate image to customers (regardless of a business unit)
	Reduction of inventory and storage maintenance costs	Accepting orders and estimating the date of implementation on a global scale
	Eliminating delays and errors in filling customers' orders	Improving financial consolidation
		Improving decision-making at the enterprise level

Source: Own work based on (Markus and Tanis, 2000 p. 180)

Companies choose to implement an enterprise system for a variety of reasons that may be technical or business-related. Both small and large companies obtain technical and strategic benefits from an investment in an enterprise system. In general, the needs and capabilities of small businesses are a subset of the needs and capabilities of large enterprises. For example, the problems of enterprises resulting from having non-integrated systems are increased in large companies by the difficulties in maintaining multiple computer systems serving the same area of the company operations. A good illustration of this mechanism is that of a large company where dozens of different general ledger packages were used during the implementation of an enterprise system and more than 20 separate computer applications to handle purchase orders (Markus and Tanis, 2000).

3.5.2. Reasons for migration to the new system version

An enterprise that uses a given enterprise system for a long time can be in a situation where it needs to update the software version, i.e. perform the so-called migration. Table 3.12 shows the causes for migration to a new system version according to the research carried out among 24 users of the *BaaN* system in Europe and North America (Kremers and van Dissel, 2000). Respondents could give multiple answers about the causes of migration to the new system and were asked to identify the key reason. In the table, the causes are divided into categories: business, technical, organisational, and environmental (external). The results illustrate that the most common cause of the system migration was the desire to have additional functionality. On the other hand, considering the categories of reasons, it appears that technical reasons were the most frequently indicated, amounting to 57% percent of the declared key reasons for migrating to the new version of the system.

Table 3.12 Reasons for migration to the new system version

Category	Reason	Key reason	Declared reason
Business	Additional functionality	29%	57%
Technical	Compliance with new standards	19%	38%
	Expiration of support for the installed version of the system	14%	24%
	Keeping the system up-to-date	14%	14%
	Disappointment with technical performance	10%	24%
Organisational	Organisational issues	10%	14%
Environmental (external)	Pressure from the value chain	5%	5%

Source: Own work based on (Kremers and van Dissel, 2000 p 55).

3.5.3. Relationships between the reasons for implementation of enterprise systems

The motivations of companies opting for the implementation of an enterprise system can form causal links. This regularity is suggested by J. Ross and M. Vitale (2000) based on research conducted in 15 U.S. companies. All surveyed companies implemented a "production control" module and various sets of other modules supporting, for example, finance, sales, and marketing. Based on interviews with respondents representing the company's top management and project management, six main motivations for enterprise system implementations in enterprises were distinguished (Ross and Vitale, 2000):

1. The need for a common system platform
2. Process improvement
3. Availability of data
4. Reducing operating costs
5. Better impact on customers
6. Improved strategic decision-making

The authors suggest that the causes are interrelated and form three groups associated with the infrastructure, capabilities and performance (Fig. 3.59). The relationship between the causes consists in the fact that the new common system platform forming the infrastructure provides new opportunities, which in turn are linked to expectations for performance improvements.

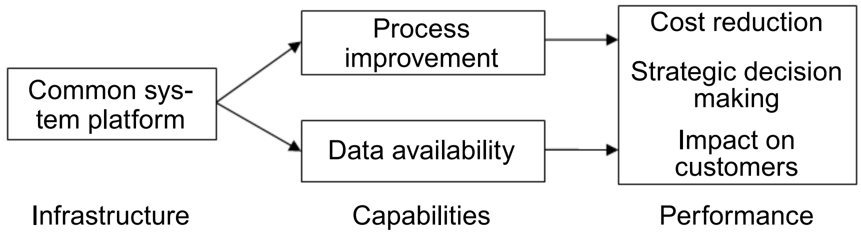

Fig. 3.59 Relationships between motivations for enterprise system implementations

Source: Own work based on (Ross and Vitale, 2000 p. 234)

3.5.4. Motivations for enterprise system implementations in the context of the effects of implementation

The motivation for the implementation of an enterprise system is a very important factor affecting especially the initial phase of implementation. However, one should ask how the initial motivations of the company transform into results achieved through implementation. This was examined by D. Chand et al. (2005) by conducting a case study at an international enterprise involved in the production of aircraft engines, which has successfully implemented the SAP system. In this study, the authors used the Kaplan-Norton Balanced Scorecard in conjunction with the Zuboff technology use level model (1985).

A modified version of the Kaplan-Norton Balanced Scorecard was used, with the following dimensions: Process, Customer, Finance, and Innovation. Referring to the motivations of business managers in the implementation of the enterprise system, the Zuboff model was used, in which the intention to use technology by enterprises can be divided into three levels: automate, informate, and transformate.

Table 3.13 illustrates the relationship of motivations to implement the system to the results achieved by specifying the assumed goals (rows labelled Purpose) and presenting the achieved benefits (rows labelled Effects). The authors consider one of three categories defined above (automate, informate, transformate) as the intention to implement the system and demonstrate the objectives and results broken down into the dimensions defined by the Balanced Scorecard.

Table 3.13 A model for linking the benefits of an enterprise system and the motivation to implement the system

	Process	Customer	Finance	Innovation
Automation / Operational benefits				
Goal	Improving process efficiency	Meeting the needs of customers more efficiently	Cost reduction	Increase in productivity
Effects	Reduction of errors; faster processing; reduced processing time; increase of throughput	Improved response time; reducing the number of complaints; reducing the number of errors	Reducing storage costs; lower labour cost	Involvement of key users in the training for operational tasks

	Process	Customer	Finance	Innovation
Informing / Tactical benefits				
Goal	Improving tactical decision making	Proactive identification and meeting of customer needs	Increase in revenue	A more efficient decision-making by employees
Effects	Better planning of work; better allocation of tasks; better access to information; better quality management; more control	Better estimation of customer expectations; increase in customer satisfaction; improved planning of tasks and deliveries	Better forecasting; market share growth	Training in the use of information and decision-making skills; empowering employees to take action
Transformation / Strategic benefits				
Goal	Adaptation to radical changes in the environment	Meeting the needs of new customers or new needs of existing customers	Improvement of market value	Routine adaptation to radical changes
Effects	Changes in technology; changes in legislation; changes in competition	Increasing the number of customers; partnership with customers	Increased capitalisation; new markets	Change management processes; broadening and deepening horizons

Source: Own work based on (Chand et al., 2005 p 568)

3.5.5. Benefits of the implementation, depending on the business or technological approach to the implementation

Motivations for companies to implement an enterprise system can be divided, in a very general terms, into technology- or business-related. Consequently, we can investigate whether the implementation was carried out from the business perspective or the technology perspective. Finally, we can ask a question how the perspective of managing the implementation affects the achieved benefits and the success of implementation. The answer to this question guided the study by O. Velcu (2007), who conducted an analysis in 8 companies in Finland using the understanding of a successful implementation based on the idea of the Kaplan-Norton Balanced Scorecard. The author proposed the so-called ERP scorecard, with the following dimensions: Changes in Business Processes, Internal Efficiency, Customers, and Finance.

Technological motivations reported by companies include: replacing the old system (the most frequent motivation cited by five of the eight companies surveyed), the year 200 problem, the need for a new enterprise system and the ease of upgrading to new versions of the system. Business motivations cited by the surveyed companies are the need for a common vision for the company, the need for a common financial strategy for the whole company, and the need for a common system with a newly acquired company.

Table 3.14 shows the similarities and differences between the implementations carried out from the business and technology perspectives, with a division of the benefits achieved defined by the ERP Balanced Scorecard. Both the implementations carried out from the business perspective, as well as the group of projects carried out from the technology perspective see the same changes in business processes involving the automation of processes and changes in responsibilities for financial management.

Table 3.14 Benefits of implementation, depending on the business or technological approach to the implementation

Implementation perspective	Type of benefit	Benefits
business	Internal efficiency	Economy of scale
	Finance	Lower overhead, sales and administrative costs Lower staff costs
business + technology	Changes in business processes	Automation of business processes Changes in the assignment of responsibility for financial management
	Internal efficiency	Shorter process execution times Transparency of the processes
	Customers	More accurate billing Improved customer service
	Finance	Margin management
technology	Internal efficiency	Shorter time of performing accounting tasks
	Customers	Faster response to changes
	Finance	Improved efficiency

Source: Own work based on (Velcu, 2007, p. 1328)

Both groups of implementations show shorter processes execution times and their greater transparency as the benefits related with internal efficiency. It is worth noticing that the projects carried out from a business perspective report the benefits related with the economy of scale, and implementations carried out from the technology perspective report shortening the length of accounting tasks.

Both groups of projects point to the benefits associated with customers in the form of better service and more accurate billing. In addition, projects carried out from the technological perspective emphasise the faster response to changes in the category of benefits associated with customers. In the category of financial benefits, both groups of projects report better margin management. Projects with an emphasis on technology emphasise the various improvements in efficiency, while the implementations carried out from the business perspective achieve benefits of reducing various costs.

3.5.6. Motivations for Polish enterprises to implement enterprise systems

The motivations for the implementation of enterprise systems for Polish enterprises are listed in the study (Soja, 2009). Based on research conducted among 63 Polish companies, 158 responses were collected on the motivations for enterprises to implement enterprise systems. The obtained data were analysed to capture similarities and differences, and consequently 10 most important motivations were distinguishes, as illustrated in Table 3.15. The percentage of answers given by respondents is assigned to each motivation.

Table 3.15 Motivations for companies for the implementation of enterprise systems according to the representatives of Polish companies

Motivation	% of responses
Access to data and better information flow	15.8%
Shortcomings of the legacy system	15.2%
Improving business performance indicators	13.9%
Improvement of business processes	12.0%
Organisational and system integration	11.4%
Development and growth of the company	10.8%
Reaction to the company environment	6.3%
Modernisation of the system	5.7%
External factors	5.7%
Implementation of corporate strategy	1.3%

Source: Own work based on (Soja, 2009)

Most frequently cited motivation by Polish respondents was providing better access to data and a better information flow. Respondents point to the need for rapid access to timely information and the need for a wider range of available data. They also point to the need for better communication and greater flow of information between employees and departments of the company.

The second most popular motivation, reported in 15.2% of the answers, concerns the shortcomings of legacy systems existing in the enterprise. The shortcomings reported include lack of functionality and issues with the performance and reliability of older systems. Respondents also point to problems resulting from lack of integration of existing system solutions in enterprises.

The third motivation is related to the desire to improve various business performance indicators. The indicators mentioned are related primarily to the quality and speed of customer service, company operating costs and reduction of inventories. Respondents also point to the time of performing different tasks and the performance of the company.

Another motivation concerns the improvement of business processes. Above all, respondents mentioned general streamlining and automating of processes taking place in different areas of company activity. They also mention the need to introduce better control of processes and departments.

Next, the enterprises were guided by the need for greater integration of the company, both an organisational and system integration. From the point of view of organisational integration, respondents mostly mentioned improving the cooperation between departments and effective exchange of information. System integration consists of the need for standardisation of processes in various departments of the company, the need for a central system and standardisation of data from different departments.

Sixth in the order of motivations is the response to the growth of the company, both in the sense of creating new branch offices and entering new markets, as well as in the sense of increasing turnover. Respondents indicate that dynamic development of an enterprise is related to the demand for rapid access to information and its processing, which requires the existence of an integrated enterprise system in the company.

Three subsequent motivations having an average popularity regard the reaction to the company environment, improving the management system and external factors. Talking about the external environment, the respondents mentioned the need for rapid response to changes taking place in the market, the need to ensure competitiveness and the need to maintain prestige among the customers. Mentioning the improvement of the company system, respondents refer to new improved system functionality and the need to replace old inefficient software. Finally, external factors are the reasons independent of the company, such as a decision of the mother company or changes in the law enforcing the introduction of new software.

Individual motivations reported by respondents concern the need for the implementation of a corporate strategy, the expected benefits from the system and solving organisational problems within the company.

3.5.7. Motivations of companies in emerging and developed economies

Table 3.16 contains a comparison of the opinion of Polish respondents with the results of existing studies, most of which were based on the experience of practitioners from developed countries. Opinions of Polish companies have been mapped to the motivations appearing in the literature, giving the percentage of respondents' answers in this study attributable to the motivations found in literature and the rank compared to the list of all motivations.

Virtually all of the motivations reported by Polish respondents were mapped to the results reported by the existing research literature. However, the question arises of the focus on individual motivations, which in many places is different depending on the level of development of the national economy, in which the company operates.

Both companies in developed economies and Polish ones agree when it comes to the most common motivation for the implementation of an enterprise system, namely the legacy systems shortcomings. Also, companies in developed countries and Polish companies indicate a desire to improve processes as one of the most important reasons. Similarly, there are also declared less important motivations associated with a response to environmental pressures and the reorganisation of the company business processes. In the latter case it is worth noting that the reorganisation of processes in Polish enterprises appears primarily in the context of greater integration of the company.

Polish companies report motivations associated with the need for greater availability of data and the adaptation to company growth more often than companies from developed countries. Polish companies are also often driven by the need to improve business performance.

Motivations which are not reported by Polish companies at all, include mainly the improvement of the decision making process and the ease of system upgrade. Both are important motivations cited by companies from developed countries. The desire to improve the IT architecture is also mentioned by companies from developed economies. The group of motivations characteristic for companies from developed countries is supplemented by those rarely mentioned by Polish companies, and often by the companies in the Western economies. These include: replacing the previous system, better impact on customers, cost reduction, and new functionality offered by the system.

Table 3.16 Mapping the results of surveys among Polish enterprises on motivations in developed countries

Rank of the importance in developed countries	Motivation	Polish respondents	
		Percentage of responses	Rank
1	Legacy systems shortcomings	15.2%	1
	Better impact on customers	4.4%	9
	Replacing the legacy system	2.5%	12
2	Reorganisation of business processes	6.3%	6
	Improved decision-making		
	Systems integration	2.5%	12
	Cost reduction	1.3%	14
	Response to environmental pressures on companies	6.3%	6
	Ease of system upgrade		
	Process improvement	11.4%	3
	New functionality offered by the system	3.2%	11
3	Improving company performance	8.9%	5
4	Externally imposed	5.7%	8
	Data clean up/ standardisation	4.4%	9
	Availability of data/ information	14.6%	2
	Improving the IT architecture		
	Need for a common vision and strategy	1.3%	14
	Adaptation to the company's growth	10.8%	4

Source: Own work based on (Soja, 2009)

Results from studies conducted among Polish practitioners suggest that Polish companies are at the stage of using enterprise systems for automating or informing the organisation. Polish companies have not yet reached the level of enterprise system use for transforming the organisation, which is available for companies from developed economies. These companies show the intention to use enterprise systems for the reorganisation of processes in the company, transformation of the company to improve the influence on customers and more efficient decision-making. The comparison of motivations of Polish companies and companies from developed economies suggests that Polish companies are more focused on short-term, immediate effects of the system, while Western companies are looking into the future for long-term effects of introducing a system in the company.

3.6. BENEFITS OF THE ENTERPRISE SYSTEM IN AN ENTERPRISE

3.6.1. Benefits achieved by companies in developed economies

Companies can achieve very different effects as a result of the implementation of an enterprise system. These benefits are shaped differently depending on the level of development of the company, which is associated with the degree of development of economy in which the enterprise operates. The benefits achieved by companies in developed countries have been examined by the consulting company Accenture, which analysed successful implementations of an enterprise system in 163 companies from Europe, USA, and Australia. The 10 most important benefits to the company as a result of the use of an enterprise system include (Davenport et al., 2002):

1. Improving decision making by managers – managers backed by the capabilities of the system are able to make better, faster decisions that are tailored to the business strategy.
2. Improved financial management – directors may provide better control of finances, better predict the financial capacity and the consequences of changes in the company operations.
3. Better customer service – access to integrated customer information enables faster and more effective implementation of their needs, resulting in higher customer satisfaction and loyalty.
4. Ease of growth and greater flexibility – an enterprise system enables easier integration with newly acquired businesses.

5. Faster and more accurate transactions – integrated and accurate databases help to reduce IT costs and improve data quality, an effect that allows the achievement of other objectives of the company.
6. Workforce reduction – achieved through more efficient service operations, leading ultimately to reducing costs.
7. Shortening of the cycles of production, customer service, etc. – the company can reduce costs and better adapt to the needs of its customers and employees.
8. Improving inventories management and asset management – an enterprise system can reduce costs through more efficient inventory management and asset management in the supply chain.
9. Reducing resources / better logistics – as the result of which the company achieves a better operational efficiency and cost reduction.
10. Increase in revenues – an enterprise system enables companies to offer new products or exploit new distribution channels, thus creating new opportunities for income generation.

3.6.2. The occurrence of the benefits over time

The positive effects of the implementation of an enterprise system do not take place immediately, their emergence usually requires time. What's more, during the implementation of an enterprise system, companies typically experience a significant decrease in performance. This decrease usually occurs within 4 – 12 months after the launching of the system (Ross and Vitale, 2000). After the implementation stage, the company which carried out a correct implementation of the system is improving its operations in subsequent stages, which can be described as stabilisation, continuous improvement, and transformation (Fig. 3.60).

The phenomenon of the apparent decline in performance after launching the new system can be explained by the need for learning by the organisation, and by the time required by employees to understand their role in the new processes. The key issues affecting the achievement of the benefits from an enterprise system include (Ross and Vitale, 2000):

- Matching the system and the enterprise (system selection, configuration, matching business processes)
- Knowledge of the users and participants of the implementation
- The process of implementation (project management, decisions on the adopted model, etc.)
- Change management

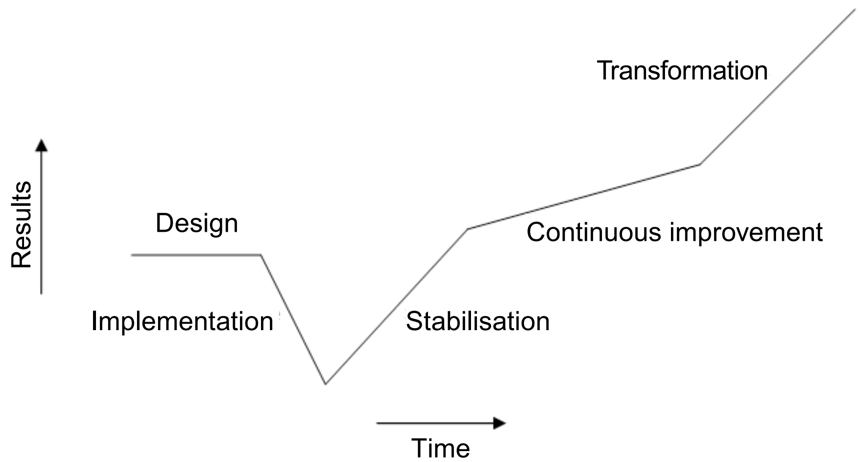

Fig. 3.60 The results of the implementation of an enterprise system over time
Source: Own work based on (Ross and Vitale, 2000 p. 236)

3.6.3. Benefits dimensions

The benefits achieved by companies as a result of the implementation of an enterprise system can be very diverse, which is related to the fact that the enterprise system penetrates deep into the structure of the company affecting the character of processes at the company and redefining job positions. One of the most important divisions of the benefits is the categorisation proposed by S. Shang and P. Seddon (2002) based on an analysis of cases and interviews with practitioners. This division includes 5 benefits dimensions, divided into 21 subdimensions, which are discussed below.

Operational benefits refer to the daily activities associated with acquiring and using a variety of resources. These activities are usually done periodically: at daily, weekly, or monthly intervals. Operational benefits include the following elements:

- Cost reduction
- Shorter cycles (e.g. sales, production, debt collection, etc.)
- Increased productivity
- Increased quality
- Improved customer service

Managerial benefits refer to management activities associated with the allocation and control of company resources, monitoring the company's operational activities and support for strategic business decisions. Managerial benefits include:

- Better resource management
- Better planning and decision making
- Improved business performance and productivity

Strategic benefits refer to activities related to long-term planning and related decisions, such as business alliances, market competition, product planning, customer retention and attracting capital. Strategic benefits include the following elements:

- Support for business growth and enabling global expansion
- Facilitating connections (alliances) with other companies
- Building business innovation
- Building cost leadership
- Generating product differentiation
- Building external relationships with customers and suppliers

IT infrastructure benefits refer to IT resources shared by the entire company, which are the basis for existing and future business applications. Benefits of this type include the following elements:

- Building business flexibility for current and future changes
- IT cost reduction
- Increased IT infrastructure capability

Table 3.17 Benefits achieved from the implementation of an enterprise system by type

Type of benefit	% of companies
Operational	73%
Managerial	55%
Strategic	56%
Related to the IT infrastructure	83%
Organisational	14%

Source: Own work based on (Shang and Seddon, 2002 p. 285)

Organisational benefits occur when using the enterprise system results in positive effects associated with the learning of the organisation, its greater consistency and focus on core activities, and the implementation of selected strategies. Organisational benefits include the following elements:

- Support for organisational change and work method change
- Better acquisition of knowledge by employees
- Greater involvement of employees
- Building a common vision of the company

- Changing the behaviour and priorities of employees
- Greater employee satisfaction

The benefits have varying intensity in companies. Table 3.17 shows the percentage of the analysed companies where various types of benefits were observed.

3.6.4. Benefits achieved by Polish enterprises

The benefits achieved by Polish companies from the implementation of an enterprise system are well illustrated by the results of research conducted among enterprises implementing an enterprise system and among experts and consultants representing the provider of a system and implementation services (Soja, 2006a). The benefits are grouped into technical, economic, organisational, and social ones and are discussed below.

Economic benefits are the most popular group of benefits, representing 38% of all reported results. They relate to the support for activities allowing for the improvement of economic performance and include the following elements:

- Inventory reduction – such as the liquidation of inventories of technical materials
- Support for financial activities – improved cost accounting and control, introduction of budgeting, full insight into the current financial situation
- Improvement of economic indicators – including shorter order delivery times, improved inventory turnover, increased sales, improved cash flow
- Reduction in employment – resulting in reduced expenditure on wages
- Enterprise control – full control over the functioning of the whole company, control over the economic condition of the company
- Cost reduction – through better production planning and accurate data analysis, reduction of staff costs
- Sales improvement – improving the quality of customer service, greater efficiency of customer service, communication over the Internet
- Greater work efficiency – better use of employees, less work on certain tasks

Technical benefits represent 30% of the declared results, refer to the computing infrastructure, and address issues relating to information processing speed and the format of information. Technical benefits include the following elements:

- Availability of information – quick access to information, a very wide range of available information, complete information in one place
- Quality of the information – data organisation, accurate and reliable data in the system, greater level of detail
- Streamlining of reporting – expanded and standardised reporting, speed of obtaining reports, possibility to create own reports

- Modernising the IT infrastructure – hardware modernisation, greater functionality of the system, standardised IT environment
- Systems integration – integration of the system with local branch systems, automatic connection of modules
- New functionalities – new opportunities due to the introduction of the system, e.g. the use of MRP II method, tax optimisation, simulation of result on a statistical basis
- Reduced labour-intensity – automation of work, calculation speed, reducing the analysis time

Organisational benefits account for 27% of the declared effects and consist of streamlining the organisational structure or to facilitating the shift towards centralisation or decentralisation. These include the following elements:

- Organisational changes – positive organisational changes, introduction of process orientation and of new processes
- Company organisation – organisation of processes and procedures, defining processes, stabilising the organisation, unification of standards and concepts
- Improving the performance of the company – better cooperation between the company departments, improving management, streamlining processes
- Better communication – improving the flow of information, streamlining document workflow
- Support for logistic processes - production control, greater flexibility in production, purchasing control
- Greater company flexibility – faster analysis of the situation, faster response to customer needs

Social benefits are by far the least reported category, representing only 5% of the results. They concern the impact of the implementation of an enterprise system on the company's employees and their attitudes. Specific effects include:

- Education of employees – a greater level of IT literacy, an increase in awareness and understanding of the processes in the company, employee development
- Employee attitudes – greater employees responsibility, greater motivation for the tasks in implementation work, a broader conversation about the problems

3.7. DETERMINANTS OF THE IMPLEMENTATION PROCESS

3.7.1. Implementation participants

An enterprise system implementation project involves many people, and teamwork (and its organisation) becomes an important issue in the deployment of enterprise systems. People involved in the project represent all levels of management in the enterprise, as well as include representatives from outside companies involved, such as the supplier of the system and the implementation services. Participants in the project implementation are grouped in the appropriate work groups created for the needs of the implementation. In the model project implementation, several groups and roles engaging the participants in this project can be distinguished (see Soja, 2001).

*Steering Committee – m*akes strategic decisions regarding the implementation, approves plans for the subsequent stages of implementation, evaluates the implementation phases of the project and allocates funds and resources related to the implementation. The Committee's tasks also include the supervision of external consultants. The Steering Committee should include the company board members, directors representing the involved departments and the Project Manager. Additionally, the Committee may include a representative (Head of the Project) of the company providing the system and providing consulting services for the system implementation.

Project Sponsor – is a member of the Steering Committee, which has the greatest responsibility for project implementation. The sponsor's task is to be the leader in the implementation and manage the Steering Committee. Ideally, the Project Sponsor is the president or executive director of the company.

Project Manager – manages operations of the Implementation Team and leads the implementation at the operational level. The Project Manager should be an employee of the company, not a hired external expert or consultant, and should be seconded to the implementation work on the basis of a full-time employment. It is important for the Project Manager to have interpersonal skills and be trusted by the employees. The Project Manager acts as a link between the Implementation Team and the Steering Committee. The task of the Project Manager is to develop a schedule of implementation work, monitor the progress of implementation in all sectors of the company, report serious problems to the Steering Committee, as well as to closely cooperate with external consultants.

Implementation Team – is a group of people responsible for implementing the system at the operational level. The Implementation Team should have a balanced composition i.e., it should include the representatives of the IT Department and other departments of the company (preferably the operating managers), as well as consultants representing the supplier of the system and implementation services. Members of the Team divide their time for the work on the implementation project and their current responsibilities. The purpose of the Implementation Team is to create an implementation schedule and report on its progress, identify obstacles and problems and take steps to address them, as well as to perform any action to assist in the fast and effective implementation of the system at the operational level.

Head of the Project (the supplier of the system) – is a person responsible for performing the tasks incumbent on the company supporting the implementation. His/her task is to participate in the development of the organisation of the project and develop a schedule for implementing the various stages, coordinate training and track the progress of the implementation. The Head of the Project also organises help and support in the event of problems with implementation and functioning of the system.

The diversity of participants in the enterprise system implementation project is a very important determinant, which significantly affects the organisation of the implementation. Companies choosing the candidates for the required roles in the implementation project are forced to estimate the willingness of employees, their skills, knowledge, and experience. Errors made at this stage can have far-reaching negative consequences for the entire implementation. For this reason it is often stated that the use of an enterprise system in a company primarily concerns the people, and to a lesser extent, the processes or technologies (Bing et al., 1999).

3.7.2. Problems encountered during the implementation of enterprise systems

Classification of problems

Implementation of an enterprise system is associated with a variety of problems, which mainly relate with the participants in the project, the company and the system used. The variety of problems is well illustrated by the results of research carried out among 65 Polish enterprises implementing an enterprise system. The results suggest the following categories of problems, listed in order from the most signficant (Soja and Paliwoda-Pękosz, 2009).

1. *Employees* – includes issues related to employees, such as fear, reluctance, lack of ability to operate the system, old habits, lack of knowledge and skills, lack of acceptance of the system, mistakes in the operation of the system.

2. *Enterprise* – issues related to the company concern difficulties in carrying out the changes, problems with the definition of the project, financial difficulties of the company, poor preparation of the company and its incorrect organisational structure, cooperation with the supplier, lack of experience of the employees, poor definition of requirements.
3. *System* – problems include system errors, poor communication between modules, efficiency, excessive complexity.
4. *IT Infrastructure* – problems with the network infrastructure, inadequate equipment (e.g. printers).
5. *System misfit* – problems with adapting the system to fit the needs of the company, lack of the required functions, problems with customisation, inadequate document templates.
6. *System replacement* – problems with the smooth transition from the legacy system to the new one, such as those related to importing data and the poor state of the existing systems.
7. *Training* – problems with training, their inadequate scope and date of organisation, problems with cooperation with the training supplier.
8. *Implementation process* – problems with the implementation of the project, its inadequate duration (usually too short), problems with employees, the project manager, and the definition of the project.
9. *System vendor* – problems with the provider of implementation service consisting in the lack of sufficient resources (especially time) and inadequate knowledge of consultants.

Problems depending on the phase of the life cycle of the system

Implementation of an enterprise system is a multi-step and often lengthy process. Depending on the phase of the implementation project, there are a variety of difficulties, as illustrated in the research by M. L. Markus et al. conducted among the experts and representatives of companies from Europe and North America (Markus et al., 2000).

The main problems identified in the organisational stage (chartering) include the following:

- Lack of focus on business results
- Company culture resistant to change
- Lack of top management support

The design stage is followed by the project (implementation) stage, during which the system is set up and launched throughout the company. At this stage, the following problems may arise:

- Software modifications

- Problems with systems integration
- Problems with product and implementation consultants
- Turnover of the project participants

The next stage is reorganisation (shakedown), during which the company makes the transition from the launch of the system to normal operation. It is associated with the following difficulties:

- Approaching the enterprise system implementation from the functional perspective
- Inappropriate limitation of the scope of the project
- Limiting training for system users
- Inadequate testing (of interfaces, modifications, integrations, exceptions)
- Lack of prior business process improvements
- Underestimation of the problem of data quality and reporting requirements

The last stage is the continued use of the system (onward and upward), during which the company gains most of the benefits of the system and is planning further steps in its development. The following difficulties can be encountered here:

- Unknown business results
- Disappointing results
- Problems with staff familiar with the system
- Problems with the migration of the system

Causal links between problems and source problems

Problems during the project implementation are complex and can affect one another. Awareness of the causal link between the problems makes it much easier to discover which problems can cause other difficulties and to focus on the real causes of problems rather than the elimination of symptoms. Fig. 3.61 shows the relationship between categories of problems (Soja and Paliwoda-Pękosz, 2009).

The suggested relationships between the problems and the further analysis of data led to the formulation of specific source problems, which include the following difficulties (Soja and Paliwoda-Pękosz, 2009):

- Knowledge of employees at different levels of the organisational structure
- Finances revealing the need to properly assess the costs of implementing the new system
- Company structure
- Changes in the company
- IT infrastructure (network, inadequate hardware)
- Training time
- Data import and legacy systems

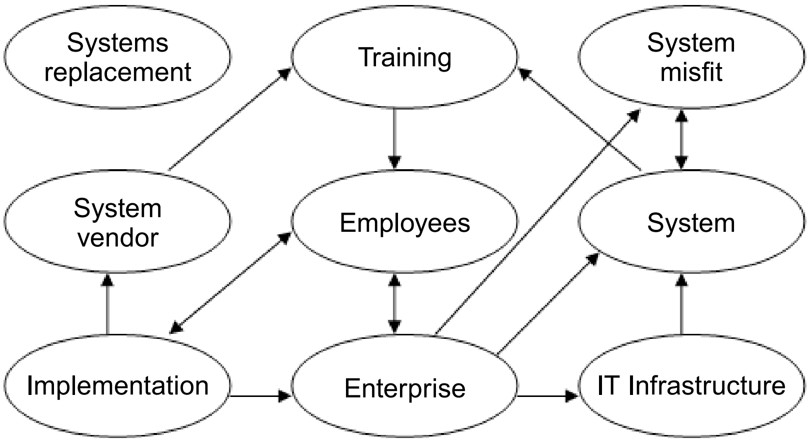

Fig. 3.61 Relationships between categories of problems occurring during the implementation of enterprise systems
Source: Own work based on (Soja and Paliwoda-Pękosz, 2009 p. 617)

3.7.3. Barriers and risk factors in the implementation of enterprise systems

Barriers occurring during implementations in Polish enterprises

The barriers experienced by Polish companies have been illustrated in the study carried out among 63 enterprise system implementation projects. The most significant barriers include the following (Soja, 2008a):

- Infrastructure – inadequate equipment, poor network infrastructure
- Time – usually limited and too short, no employees time and dedicating too little time for training
- Knowledge – lack of computer skills and knowledge regarding the enterprise system, lack of competence of employees
- Finances – a financial barrier caused by high costs of the system and training
- Reluctance – the reluctance of employees to changes, the system, learning and using the system
- Resistance – the resistance of employees and users of the system in relation to the system and changes
- Labour-intensity – labour-intensity of the implementation project caused by the necessity of learning, documenting the implementation tasks, working in parallel in two systems
- Cost – the overall high cost of implementation, high cost of the system itself

- Training – poor quality of training
- Fear – employees' fear regarding the new system and changes, fear of failure
- Attitudes – negative attitudes of people, lack of confidence in the system, lack of acceptance of the system
- System fit – incorrect system fit to the legal requirements, lack of fit to existing systems, incorrect document types
- Habits – employees are accustomed to the current style of working and the existing systems

Causes of failures of enterprise system implementations

E.J. Umble and M.M. Umble (2002), when analysing the reasons for the failure of ERP system implementation projects, came to the conclusion that there are ten reasons for failure of the enterprise system implementations (Soja, 2008b):

1. Lack of leadership from top management – the management board does not participate actively in the project and does not carry out changes in the company required by the enterprise system, does not consider the implementation of the enterprise system as a transformation of the company.
2. Automating existing ineffective processes in the new system – the system often requires a different organisation of the company, and existing processes should be reorganised under the dictation of the enterprise system.
3. Unrealistic expectations – companies do not appreciate the enormity of resources needed to implement the system, employees expect the company's performance will be improved immediately, whereas the company should be prepared for an initial decrease in performance.
4. Poor project management – often due to being surprised by the scope, size and complexity of the implementation, resulting in the lack of proper planning and control in project management.
5. Inadequate training – top managers and system users must be trained. It is necessary to understand how the enterprise system is integrated with the activities of the entire company.
6. Maintaining the status quo – people's fear of the changes introduced by the enterprise system, which may relate to job loss, deterioration of working conditions, and loss of position and importance within the company. Uncertainty in the use of the new system. People fear of whether they will manage the new situation.
7. Bad fit – enterprise system capabilities do not fit into existing business processes and business needs. A significant mismatch can cause general chaos, and a small mismatch causes serious problems for the implementing parties and users.

8. Inaccurate data – causing a "domino effect" and errors in the planning, orders, etc. In effect, there is less trust in the system.
9. Perception of the enterprise system implementation as an IT project – resulting in an inability to realise the full potential of the system. There is no reorganisation of the company and adaptation of the system to the organisation.
10. Significant technical problems – system errors, problems with interfaces, hardware problems, which unresolved and neglected can lead to the failure of the implementation.

Risk factors in the implementations of enterprise systems

Risk factors are conditions whose occurrence exposes the project to a greater likelihood of failure (Soja, 2008b). In the context of the implementation of enterprise systems, the most important risk factors include the following (Sumner, 2000; Huang et al., 2004):

- Not carrying out changes in business processes in order to adapt the company to the system
- Lack of appropriate governance structure and effective project management methodology
- Lack of commitment of top management to the project
- Insufficient training
- Insufficient knowledge within the organisation
- Ineffective communication
- Conflicts between company departments
- Inadequate implementation team

3.7.4. Success factors in the implementation of enterprise systems

The concept of success factors in the implementation of enterprise systems

The concept of critical success factors, originally called success factors, was first introduced to the literature on management by D. Ronald Daniel in 1961 (Rockart, 1979, p. 85). However, J. F. Rockart (1979) was the first researcher who used the concept of critical success factors in the area of information systems to measure the information needs of managers of the company.

Critical success factors represent areas of activity of companies in which obtaining satisfactory results ensures good performance and competitiveness of the entire organisation. In other words, this is a limited number of key areas of company activity in which achieving good performance is necessary for the results to be satisfactory throughout the organisation. If the results in key areas are not satisfactory, then the results of the entire organisation are below expectations (Rockart, 1979).

As a result, critical success factors are areas of activity that should be subject to constant observation by the managers. There should also be a way to measure the factors, while the measurement itself should be performed continuously. Success factors support the attainment of the objectives that the organization intends to achieve. Therefore, they represent the areas of activity in which achieving good results is necessary to achieving the intended objectives of the organisation. Thus, the success of the organisation is seen as achieving the desired objectives.

The issue of critical success factors is applicable not only within business operations. It can be also applied to various projects and ventures. In the context of projects aimed at implementing an enterprise system, critical success factors include areas, circumstances, mechanisms, etc. which are key to the success of the project.

Discussion of selected key models of success factors

The study of ERP implementations using an approach based on critical success factors started in the late 90s of the twentieth century (Bing et al., 1999; Holland and Light, 1999). Within the next few years, the researchers have presented several models of critical success factors in implementations of enterprise systems (Al-Mashari et al., 2003; Brown and Vessey, 2003; Nah et al., 2001; Nah et al., 2003; Somers and Nelson, 2004; Sumer, 1999). One reason for the multiplicity and variety of critical success factors is the fact that these factors are dynamic and change over time (Pinto and Prescott, 1988; Rockart, 1979).

Factors proposed in the literature represent various aspects of implementations and different levels of generality. There are models containing five factors only (Brown and Vessey, 2003), as well as proposals containing over 20 elements (Soja, 2004; Somers and Nelson, 2004). Moreover, existing models of success factors use a variety of divisions of factors into categories (e.g., tactical, strategic, organisational, technical), and there is no generally accepted way to group elements into coherent categories of similar factors.

The multiplicity of circumstances and factors that potentially determine the success of implementations is presented in the model defined in (Soja, 2004, 2006d). This model is based on the analysis of existing proposals of factors and experiences in business practice. It contains 26 success factors grouped into five categories. The separate categories include factors related to the participants in the implementation, involvement of top management, definition and organisation of the project, project status and information systems (Soja, 2004, 2006d). Factors associated with the participants of the implementation project include:

- Project manager – project manager who is a person from within the company, spending most of his time on the implementation
- Team composition – an implementation team consisting of a variety of people with high skills and substantive knowledge about the company
- Team involvement – involving the project manager and team members in the implementation project for a significant amount of time
- Motivation system – the existence of the incentive system rewarding participation in the implementation project and adherence to deadlines
- Cooperation with the supplier – good cooperation with the supplier of the enterprise system, its relevant competences and high level of service provided

The factors that define top management's commitment include the following elements:

- Top management support – the support of top management for the implementation project and active involvement in the implementation work
- Top management awareness – in regard to the goals, complexity and labour-intensity of the project, the existing limitations, the required expenditures and the project inevitability
- Top management participation – in the definition of the implementation goals and development of its schedule

Factors related to the definition and organisation of the implementation include the following elements:

- Linking with the strategy – linking the implementation with the corporate strategy, treating it as a measure for achieving the company's strategic objectives
- Implementation goals – defining the objectives of the implementation project at the enterprise level as a whole and given in economic categories
- Detailed schedule – defining the detailed scope of implementation, detailed plan and timetable with the assignment of responsibilities
- Pre-implementation analysis – analysis and diagnostic review of the company prior to the implementation work and creating a model of the company operation with the support of an enterprise system
- Organisational changes – conducting organisational changes in the company and its business processes
- Monitoring and feedback – exchange of information between the implementation team and the users of the system
- Promotion of the implementation – notification of the other company staff about the project by the implementation team
- Fast effects – visible fast partial results of the implementation
- Appropriate training – an adequate training programme adapted to the needs of the company

Factors associated with the status of the implementation project include the following elements:

- Investment plans – formal introduction of the implementation project into the company's investment plans
- Project team empowerment – giving appropriate decision-making powers to the members of the implementation team and their high placement in the hierarchy of the company
- Financial resources – ensuring the financial resources for the implementation work
- Working time – providing adequate working time for the implementation teams
- IT Infrastructure – providing an appropriate system and hardware infrastructure for the implementation

Factors related to information systems are the last group, and include the following elements:

- System reliability – the reliability of the system, its user-friendliness and adaptation to the needs of the company
- Minimal customization – the use of ready models and solutions offered by the system
- Legacy systems – smooth adaptation of the legacy systems to the operation in the enterprise system environment
- Implementation experience – experience of implementing information systems gained in the past

Studies conducted among Polish companies illustrate that the factors which are the most influential to the success of implementation, regardless of the type of an implementation project, are (Soja, 2006d): system reliability, implementation team involvement, composition of the implementation team, working time, cooperation with the supplier, detailed schedule, and top management support. However, this impact is small and the results suggest that the impact of factors on the success of the implementation should be considered while taking into account the division of implementations in terms of duration, implemented scope and size of the enterprise.

The impact of certain factors in these divisions becomes apparent. The success of implementations in small companies is most determined by the implementation experience, but it has no meaning in the case of large enterprises, where a detailed schedule is the most decisive factor. In implementations with a full scope of system functionality, the reliability of the system has the greatest impact on the success, but in partial implementations it is financial resources. In long projects, the infrastructure has the greatest impact on success, while linking to the strategy supports success in short ones.

The participants of the survey do not appreciate the significance of certain factors, which in light of the research have a significant impact on the success of implementations. The factors underestimated by experts include system reliability and cooperation with the supplier, while the representatives of the companies do not appreciate the importance of the work schedule. All of the respondents did not consider the implementation experience and relation to the strategy important. Finally, the factor considered by experts to be one of the most important ones, associated with the project manager, which also has a very high level of occurrence in the examined implementations, has no impact on the success of implementations.

Success factors typical for Polish companies and their comparison with developed countries

The Polish economy is an example of emerging economy in a transition (Roztocki and Weistroffer, 2008). Conditions for enterprise system implementation projects conducted in Poland differ somewhat from those taking place in developed countries. This was well illustrated in the study (Soja, 2010b), which is based on the opinion of 164 participants of enterprise system implementations. Table 3.18 shows the comparison of the validity of the critical success factors found with the factors common in developed countries.

Participants of enterprise system implementations carried out in Poland agree with the practitioners from the developed countries in the evaluation of several crucial factors, which include: support of top management, change management and composition of the implementation team. By contrast, Polish practitioners evaluate as more important, compared to the representatives of developed countries, the factor associated with motivation and morale of the implementation team. Two other factors rated higher by the Polish respondents are the selection of consultants and the selection of the enterprise system. Speaking of the implementation team, it is worth mentioning that Polish respondents agree with the results of prior research concerning the great importance of the proper composition of the implementation team, but put a lot more weight to the operation of the team. On the other hand, totally missing in the opinions of Polish respondents is the problem of the so-called balanced composition of the implementation team. This factor opens up a group of issues absent among Polish respondents.

The factors neglected by the Polish practitioners include, besides the balanced composition of the implementation team, the issues related to the IT infrastructure, managing cultural change and the post-implementation evaluation. This group is supplemented by the need for changes in the company, which is given little importance by Polish respondents, but has one of the leading positions in the ranking according to the literature.

Table 3.18 Comparison of critical success factors found in the Polish implementations with those in developed countries

Critical success factor	Polish implementations	Developed countries
Support of top management	***	***
Selection of consultants	***	**
Motivation and morale of the implementation team	***	*
Change management	***	***
Implementation team: the best people	***	***
System selection	**	*
Training	**	***
Implementation strategy	**	**
Project vision	*	**
Communication plan	*	*
Reorganisation of processes and system configuration	*	***
Balanced composition of the implementation team		**
IT infrastructure		*
Managing cultural change		*
Post-implementation evaluation		*

Source: Own work based on (Soja, 2010b)

The above-mentioned table partially confirms the results of research conducted in developing countries, but also provides some interesting insights. One factor that is critical regardless of the level of economic development is the support of top management (Ngai et al., 2008). The issue which is typical for developing countries and seems to characterise the projects analysed in Poland is weak emphasis on business process reengineering and lack of experience in the process management (Huang and Palvia, 2001). To some extent, project management is also underestimated, which partly characterises the implementation projects in developing countries (Ngai et al., 2008). By contrast, two other issues characterising implementation projects in developing countries, poor IT infrastructure and maturity, and problems with financial resources, usually due to the small size of companies, generally do not seem critical for the studied Polish enterprises. Finally, the respondents of this study did not consider relevant the issues related to cultural differences, which are typical for Far East Asian developing countries (Davison, 2002).

3.8. THE EVALUATION OF THE IMPLEMENTATION OF AN ENTERPRISE SYSTEM

3.8.1. The need for multidimensional and multi-stage evaluation of IT projects

Evaluation of the enterprise system is a complicated topic, which consists of the complexity and specificity of the system itself, as well as the organisational conditions of the implementation process. Enterprise system implementation projects should be considered both as organisational projects and as the ones related to the IT area. The presence and mutual interweaving of these two areas creates the need for a proper assessment of project implementation, both as projects changing the organisation of the company and as IT projects.

Evaluation of advanced IT projects, which enterprise system implementations undoubtedly are, should be multidimensional and should be multi-stage. A good illustration of the requirements that need to be put in relation to the comprehensive assessment of IT projects is the proposal of the eight-stage evaluation formulated by G. Fitzgerald (1998), which includes the following steps:

- Estimating project costs
- Assessment of what is the project's contribution to business strategy
- Identification and analysis of the benefits of the project
- Analysis of the side effects of the project
- Estimating the flexibility of the project
- Assessment of the extent to which the project is practical and easy to implement
- Assessment of project risks
- Conducting a practical trial to test the assumptions and the expected benefits (e.g. conducting market research, creating a prototype)

3.8.2. Assessment criteria required by the specificity of the enterprise system

The above-mentioned requirements for a multi-dimensional evaluation of IT projects also illustrate the complexity of the process of enterprise system assessment, because estimation of a system of this class struggles with all the problems characterising IT projects, but also has to deal with all the issues typical of enterprise system implementation projects. An effective method for evaluating the enterprise system should have several properties that will ensure its correct operation and compliance with all requirements imposed by the context of the enterprise system. The most important properties which the method of evaluation of an enterprise system should be characterised by are discussed below (Sneller and Bots, 2005; Soja, 2006c).

- *Taking into account the specificity of the enterprise systems* – the method should take into account business process reengineering (BPR), and should refer to the model solutions based on best business practices contained in the system. These two features are typical of enterprise systems and are not found in most of the methods used to evaluate IT investments.
- *Completeness* – the method should be complete, meaning it should cover all phases of the enterprise system project life cycle and all associated costs and benefits.
- *Financial estimate* – the method should provide support for the evaluation in financial terms.
- *Risk management* – the method should contain mechanisms for risk management during implementation. The most important risk factors include: the excessive cost of implementation (Stefanou, 2001), lack of benefits because of poor match between the system and company processes (Sumner, 2000), no improvement in the situation of companies involved in enterprise system implementation being in a bad financial condition (Hunton et al., 2003) and operational problems after the launching of the system (Markus et al., 2000).
- *Taking into account the size of the company* – the method should take into account the size of the company and should be suitable for use not only in large businesses, but also in small and medium companies.

3.8.3. Selected approaches to the evaluation of an enterprise system

There are several approaches to the evaluation of an enterprise system, proposed by the researchers studying enterprise system implementations, the implementation service providers, and system manufacturers. Suppliers of integrated packages primarily focus on supporting the actual implementation process and provide tools mainly for the selection and configuration of business processes based on solutions predefined in the system, called reference models (Kumar and van Hillegersberg, 2000). Examples of available software tools of this type are Business Navigator and R/3 Analyzer from SAP (Kale 2001) and the Dynamic Enterprise Modelling application offered in the Orgware package that supports the implementation of the Baan system (Perreault and Vlasic, 1999).

Manufacturers of enterprise systems offer solutions primarily related to the system packages proposed by them, and usually it is difficult to obtain specific information on the evaluation methods that were used in the tools provided (Sneller and Bots, 2005). However, the most interesting in the practical application is the evaluation method which is independent of any particular package. Using such a method enables to compare the offers of different enterprise system providers and can be a useful tool in the process of selecting the system.

The methods which are independent of system solutions include the proposal of C. Stefanou (2001) which contains a general framework for the ex-ante evaluation of an enterprise system. The author stresses the importance of the system life cycle and proposes a number of evaluation steps, which include (Stefanou, 2001):

- Clarification of the business vision
- Comparison of the company's business needs with available opportunities and existing limitations (technical, organisational, people-related, financial and time-related)
- Evaluation of the enterprise system package, its supplier and implementation partner
- Evaluation of the implementation project focusing primarily on the costs and benefits associated with implementing an enterprise system
- Assessment of the post-implementation stage regarding the costs and the benefits associated with the future operation, maintenance and extension of the system with additional functionality

The method proposed by A. Teltumbde (2000) is an application, in the context of enterprise systems, of the previously discussed general method of evaluation of IT projects presented by G. Fitzgerald (1998). Teltumbde expanded Fitzgerald's approach primarily by the evaluation of the system supplier reliability and eventually presented 10 evaluation criteria and several stages during which the assessment of the system should be carried out. The proposed criteria include the following issues (Teltumbde, 2000):

- Fit with the corporate strategy
- Technological advancement of the system
- Change management
- Risk
- Ease of implementation
- Fit with the business processes in the company
- References of the system provider
- System flexibility
- Cost
- Benefits

Evaluation of the enterprise system should take place in several stages, which include (Teltumbde, 2000):

- Creating the organisational infrastructure for the evaluation
- Selecting a set of system solutions for the evaluation
- Preparatory phase
- Context setting phase
- Evaluation and selection phase
- Approving the choice
- Review and possible improvement of the evaluation

3.8.4. ABCD assessment method and the MRP II system implementation class

Evaluation of the effectiveness of the MRP II system implementation is a very important issue for the company management wishing to consciously manage the implementation process. Assessment of the implementation of an MRP II system is particularly important in the context of the existing profound differences in the use of MRP II packages by different companies (Wallace, 1990, p. 8).

In order to support the assessment of the effects of the MRP II package, an implementation evaluation method was developed – the so-called ABCD method. Its author, Oliver Wight, is one of the founders and popularisers of MRP and MRP II standards. Assuming that the evaluation of the effectiveness of business processes should be continuous and cyclical, Wight came to the conclusion that it must be in the form of self-evaluation. The ABCD method is intended for companies wishing to evaluate their own business processes in comparison to the so-called best practices. A group of processes is considered as these practices, having an essential significance for the efficient operation of the company, and which have been identified through analysis of hundreds of U.S. companies (Popończyk, 1997, p. 32).

Four classes of implementation have been defined: A, B, C, and D, as well as a list of dozens of detailed questions, answering which enables companies to make a self-assessment. The questions are divided into seven groups: (1) strategic planning, (2) operational planning, (3) operations control, (4) data management, (5) performance measurement, (6) documentation, (7) training. The main issues raised in each group are discussed below (Czajkiewicz, 1998, p. 62, Soja, 2001).

Strategic planning. A document in the form of a business plan should be created as a result of strategic planning. It should be adopted and accepted by employees and managers of the company. A review of the plan should take place every quarter. Sales forecasts should be grouped by production processes, not market categories. Forecasts should be updated before fixing the current production plan. Production planning should be done every month with the participation of the managers of the technological, financial, manufacturing, materials and sales departments.

Operational planning. Requirements for operational planning are related primarily to the development and implementation of production schedules. They should be prepared weekly and should be closely linked to the production plan. The dates in the schedules should be precisely complied with. Production capacity should be planned taking into account predicted changes of available resources.

Operations control. During the control of operations, particular attention is paid to the improvement of processes. The plan for the improvement of company performance should be documented, and products and processes should be rationalised in terms of benefits for the company. Controlled elements include: safety stock, supplier evaluation, and stock recording. There should be a single official supplier evaluation system based on quality, quantity, delivery time, and price. Inventorying of stocks should be continuous and stock record accuracy should be equal to 99%.

Data management. Proper data management requires the existence of a formal structure for the products (bill of materials – BOM) covered in the production process. Each organisational unit of the company should work on the basis of one common BOM, which should be updated by the control system for structure changes.

Performance measurement. Measuring the performance is used to accurately determine the efficiency of the implementation of the MRP II system. There are a number of requirements necessary for classifying the implementation as class A, which defines the best implementations according to the ABCD Method:

- Return on investment at 90% of the planned ROA (Return on Assets)
- Sales at 90% of forecasted sales
- Production plan completed in 95%
- Manufactured finished goods represent 95% of products included in the production schedule
- Manufacturing orders launched on time make up 95% of all launched manufacturing orders
- Actual production hours worked make up 95% of the hours defined in the schedule
- Information concerning bill of materials have an accuracy of 95%
- Stock record accuracy is 95% for each location

- Accuracy of information on manufacturing processes at 95%
- Deliveries accepted on time, production and customer orders completed on time represent 95% of the total number

Documentation. There should be sets of written intructions for top management and the individual functional areas of the company. The procedures should cover all conditions and situations which the user can encounter.

Training. Training should be treated as an ongoing educational process. From the beginning of the implementation process, the company should have educational programs on the MRP II developed.

The above-mentioned topics related to the ABCD method show what a big challenge the implementation of the MRP II system is for companies, and indicate most of the weaknesses in implementation processes. General characteristics of MRP II implementation classes are shown in Table 3.19.

Table 3.19 MRP II system implementation classes

Class	Results	Characteristics	Effect
A	90%	Fully integrated system. Top management uses a formal system for decision making.	The system is used effectively company-wide; generating significant improvements in customer service, productivity, inventory and costs.
B	80%	The formal system implemented but not all system components work effectively. Top management do not use the system actively.	The system is supported by top management; it is used by middle management in order to achieve significant improvements of company activities.
C	70%	MRP is used to generate orders but not as a system for priorities planning. Not all modules are implemented. Formal and informal systems exist.	The system is primarily used as a better method for ordering materials; it contributes to better inventory management.
D	50%	The formal system does not work or is not implemented. Lack of data integration. Low level of user confidence in the system.	The information supplied is inadequate and not understood by the users; it provides little help in running the business.

Source: Own work based on (Czajkiewicz, 1998, p. 64; Soja, 2001; Wallace, 1990, p. 9)

3.8.5. Criteria for evaluating the application of the enterprise system in an organisation

Previously discussed concepts of the enterprise system evaluation contain valuable suggestions for the issues which must be paid attention to during the effective evaluation of the system. They allow the formulation of postulates for the method of enterprise system assessment, which aims to provide a comprehensive and holistic approach to evaluating the enterprise system. A number issues and actions that should be taken into account in the comprehensive evaluation of the enterprise system are presented below (Soja, 2006c).

Company's current state – diagnosis of the current state of company, taking into account the existing strategy, organisational structure and business processes. Enterprise resource assessment in financial terms and the estimate of human capital. Evaluation of IT infrastructure including hardware and software owned and network and communication technologies used. As a result of the diagnosis, areas that require organisational changes and improvements in infrastructure should be identified.

Company target model – on the basis of diagnosis of the company's current state, a proposition of the company model including strategic goals, development plan for the next few years, main business processes, planned organisational structure, and the vision of the IT architecture including hardware and software. The company model developed should embrace next few years, because some effects of the implementation of an enterprise system may take place after prolonged use of the system. It seems that in most cases, the horizon of five years should be sufficient. It is recommended that the target model does not include references to any particular system, because this will allow its use in the evaluation of various enterprise systems.

Enterprise system – evaluation of enterprise system package quality, which should take into account a whole range of issues, including: reliability, flexibility, architecture, interfaces to other software solutions, functionality, provider's technical support, upgradeability, ease of implementation, and the availability of tools supporting the implementation process.

System provider – since the implementation of an enterprise system is usually carried out with support from the supplier of the system, it is very important to evaluate experiences and competences of an implementation services provider. The company's ability to provide effective support during the implementation, the availability of experienced consultants and references from implementation projects in similar companies and industries should be evaluated. Also important is the evaluation of the supplier's financial stability and prospects for its further development, since the introduction of the enterprise system often results in the company's long-term dependency from the system supplier.

System fit – estimating the extent to which enterprise system meets the requirements of the company and how the model fits the target company model. The assessment should take into account the capabilities and limitations of the system and the business solutions embedded in the system. This should be done in terms of (1) organisation fit, taking into account the planned method of conducting business, and (2) system fit, taking into account the need for integration with legacy systems. As a result, one should identify the necessary organisational changes and changes in the IT infrastructure, including, among others, identification of the applications and interfaces which should be created. The assessment of whether the system fits the needs of the organisation can include building of several scenarios depicting the operation of key business processes and illustrating the behaviour of the system on their basis (Hedman and Borell, 2004).

Company transformation – evaluation of the transition process from the actual state to the target company model. This evaluation should take into account the capabilities of the enterprise system, the company's strategic goals, the identification of necessary organisational changes and changes in IT infrastructure (hardware and software). The required modules of the enterprise system and the software applications and interfaces needed to be developed should be identified. The resulting company model should provide a vision of the company after the purchase and implementation of the enterprise system.

Implementation – estimation of the implementation process in terms of resources needed (time, cost, people). A general implementation schedule should be created that contains the allocated tasks, preliminary dates of task completion, and people responsible for tasks. Selection should be made regarding the implementation strategy and the method of launching the system that can take the form of a big-bang, phased, parallel or pilot implementation approach (Sarkis and Sundarraj, 2000; Soja, 2001).

Implementation considerations – estimation of the factors having a potential impact on the success of the project and identifying critical success factors. Particularly important is the assessment of employees' competences and their availability for implementation tasks. Based on research conducted among Polish companies implementing enterprise systems, the factors having the greatest impact on successful implementation include (Soja, 2004, 2006d): system reliability, implementation team composition and involvement, implementation schedule, cooperation with the supplier, top management support, linking with the strategy, appropriate IT infrastructure, adequate budget, and experience in complex IT/IS system implementation.

Risk – evaluation of risk connected with enterprise system implementation in the copmany, which might be connected with the enterprise system package, computer technology being used, and the implementation process (Teltumbde, 2000). At this stage, potential problems that may arise during enterprise system implementation and use should be identified and appropriate remedial measures planned.

Benefits – this is the final step in the evaluation of the enterprise system where all the anticipated benefits of using the system in the enterprise are evaluated. This evaluation should include both tangible and intangible benefits from the use of the system. In the case of the latter, steps should be taken to measure and evaluate them in financial terms (Murphy and Simon, 2002). Examples of tangible benefits include the reduction of operating costs, while the intangible benefits include organisational improvements and better access to information. The potential benefits of using the system should be evaluated for several consecutive years.

Costs – this is the second, in addition to estimating the benefits, final stage of the process of enterprise system package evaluation. It consists in evaluating all costs connected with enterprise system implementation and exploitation. The calculation must take into account the company transformation plan and the costs connected with system modification and additional software development projects. While estimating the costs of the project, several categories of costs should be considered, which can be divided into (Sarkis and Sundarraj, 2000): (1) direct project costs (related to the software, hardware, system installation and configuration, consulting, training), (2) indirect human costs (related to management staff involvement, employee time devoted to training, implementation duties and system maintenance, pay raises, staff turnover), and (3) indirect organisational costs (associated with decreased productivity, strains on company resources, restructuring the organisation). As in the case of the benefits, costs should be evaluated for several consecutive years.

3.8.6. Stages of the process of evaluating the application of an integrated system

The issues presented in the suggested process of evaluation should be assessed in an appropriate order. Several stages of the procedure including various activities can be distinguished. There are dependencies between the stages that cause these issues to form a causal link. Fig. 3.62 shows the main dependencies between issues in the evaluation process.

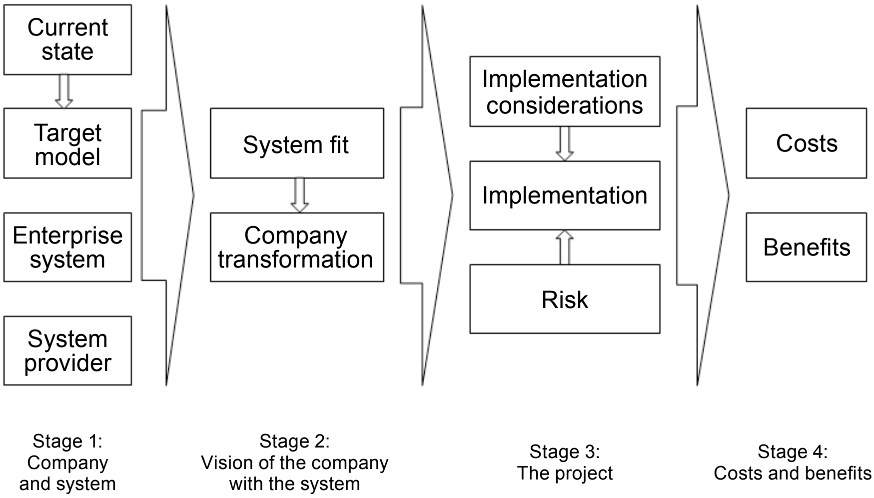

Fig. 3.62 Stages of evaluating the application of an enterprise system
Source: (Soja, 2006c)

In the model shown, subsequent stages depend on previous ones, as indicated by arrows. There are often dependencies between actions at each stage, which also is shown using arrows. Benefits and costs are estimated at the end of the process, which should provide information on the expected profitability of the introduction of an enterprise system in the company.

It should be emphasised that outlining changes in the organisation of the company and IT infrastructure is the result of many steps in the evaluation process. All these postulates should be carefully noted and be taken into consideration during the final evaluation of costs and benefits of the project.

The presented model for the evaluation of the enterprise system is discussed under the assumption that the system comes from a single supplier. However, this assumption is not necessary, the proposed model is also suitable for applications where individual modules come from different suppliers. However, we must remember that this situation greatly complicates some steps in the evaluation process, like the evaluation of system capabilities or assessing the competence of suppliers. What's more, the conditions of implementation also change and there are new issues related to adapting the system to the needs of the organisation and its maintenance (Light et al., 2001).

The proposed model contains the framework of enterprise system evaluation in the fullest perspective. Execution of all presented steps should help in completing the picture of the impact of the enterprise system on the company. However, in practical application, depending on the needs of a particular organisation, some steps may be omitted. Evaluation can also be carried out at different levels of detail, which depends on both the needs and capabilities of a given company, i.e. the allocated budget, staff capabilities, etc.

Today's enterprise systems are extremely complex software packages that include mechanisms to support management of the entire enterprise, integrating all areas of the company and have the possibility of interorganisational integration. The use of an enterprise system in the company is beneficial for many reasons, therefore more and more companies and institutions choose to take advantage of it. However, we need to realise that the actual implementation of such a system is a big challenge for the company undertaking this task.

This situation is caused by a number of factors discussed in this chapter, which include high cost of the system and its implementation, the need to adjust the system to a particular business environment, often associated with a change of business processes in the organisation, the need to involve many people in the implementation project (including stakeholders from outside the company) and time-consuming aspect of the implementation project. However, given the potential benefits from enterprise systems use and the requirements of today's economic environment, it seems that the use of an enterprise system is a necessity for companies. Therefore, the answer to the following question is extremely important and timely: How should the enterprise system be implemented to be successful?

REFERENCES

Abramowicz W., (2008), Filtrowanie informacji, Wydawnictwo Akademii Ekonomicznej w Poznaniu, Poznań

Ackoff R. L., (1967), Management Misinformation Systems, Management Science, Vol 14, No. 4, December, pp. 147–156

Ackoff R. L., (1989), From Data to Wisdom, Journal of Applied Systems Analysis, Vol. 16, pp. 3–9

Adamczewski P. (1999). Systemy zintegrowane klasy ERP – realizacja i rozwój. [in:] Kubiak, B. F., Korowicki, A. (ed.) Human – Komputer Interaction'99. Gdańsk. pp.129–139

Akkermans H., Bogerd P., Yucesan E., Van Wassenhove L. (2003) The impact of ERP on supply chain management: exploratory findings from a European Delphi Study, European Journal of Operational Research, 146 (2), 284–301

Allen T. J., Scott Morton M. S., (1994), Information Technology and the Corporation of the 1990s. Research Studies, Oxford University Press, Oxford

Al-Mashari M., Al-Mudimigh A., Zairi M. (2003) Enterprise resource planning: A taxonomy of critical factors. European Journal of Operational Research, 146, 352–364

Ashby W. R., (1957), An Introduction to Cybernetics, Chapman & Hall Ltd., London

Baker R. (1996), Modelowanie związków encji, WNT, Warszawa

Bakos J. Y., Treacy M. E. (1986), Information Technology and Corporate Strategy: A Research Perspective, MIS Quarterly, Vol. 10, No. 2, pp. 107–119

Barron I., Curnow R. (1979), The Future with Microelectronics, Francis Pinter, London

Bateson G. (1972), Steps to an Ecology of Mind. Collected Essays in Anthropology, Psychiatry, Evolution, and Epistemology, University of Chicago Press

Berger P. L., Luckmann T., (1983), Społeczne tworzenie rzeczywistości. Państwowy Instytut Wydawniczy, Warszawa

Bertalanffy von L. (1968), General System theory: Foundations, Development, Applications, George Braziller, New York

Bertalanffy von, L. (1972), The History and Status of General Systems Theory, The Academy of Management Journal, Vol. 15, No. 4, pp. 407–426

Beynon-Davies P., (1999), Inżynieria systemów informacyjnych, WNT Warszawa

Bingi P., Sharma M.K., Maneesh K., Godla J.K. (1999) Critical Issues Affecting an ERP Implementation. Information Systems Management, 16 (3), 7–14

Bocchino W. A., (1975), Systemy informacyjne zarządzania. Narzędzia i metody., WNT, Warszawa

Boehm B. (2002), Get Ready for Agile Methods, with Care. Computer, January, pp. 64–69

Bohem, B., (1986), A spiral model of software development and enhancement. ACM SIGSOFT Software Engineering Notes. Vol. 11, No. 4, August, pp. 14–24

Bond B., Genovese Y., Miklovic D., Wood N., Zrinsek B., Rayner N. (2000) ERP Is Dead – Long Live ERP II, GartnerGroup, New York, NY

Boynton A. C., Zmud R. W., (1987), Information Technology Planning in the 1990's: Directions for Practice and Research, MIS Quarterly, Vol. 11, No. 1, pp.59–71

Brown C.V., Vessey I. (2003) Managing The Next Wave of Enterprise Systems – Leveraging Lessons From ERP. MIS Quarterly Executive, 2(1), 65–77

Brown, A. E. , Grant, G. G. (2005), Framing the frameworks: A review of IT governance research, Communications of the Association for Information Systems, Vol. 15, pp. 696–712

Bubnicki Z. (1993), Podstawy informatycznych systemów zarządzania. Wydawnictwo Politechniki Wrocławskiej, Wrocław

Capurro R., Hjørland B. (2003), The concept of information, Annual Review of Information Science and Technology Vol. 37, pp. 343–411. http://www.capurro.de/infoconcept.html (retrieved: 3.07.2008)

Carr N. G., (2003), It does not matter, Harvard Business Review, Vol. 81, No.5, May, pp. 41–49

Cartlidge A., Hanna A., Rudd C., Macfarlane I., Windebank J., Rance S., (2007), An Introductory Overview of ITIL V3, The UK Chapter of the itSMF, ISBN 0-9551245-8-1, http://www.best-management-practice.com/gempdf/itSMF_An_Introductory_Overview_of_ITIL_V3.pdf

Castells M., (2007), Społeczeństwo sieci, PWN, Warszawa

Cats-Baril W. L., Jelassi T., (1994), The French Videotex System Minitel: A Successful Implementation of a National Information Technology Infrastructure, MIS Quarterly, Vol. 18, No. 1, pp. 1–20

Chand D., Hachey G., Hunton J., Owhoso V., Vasudevan S. (2005) A balanced scorecard based framework for assessing the strategic impacts of ERP systems, Computers in Industry, 56, 558–72

Checkland P., (1993), Systems Thinking, Systems Practice. John Wiley & Sons, Chichester

Checkland P., Holwell S. (1998), Information, Systems and Information Systems. John Wiley & Sons, Chichester

Chen I. (2001) Planning for ERP systems: analysis and future trend. Business Process Management Journal, 7 (5), 374–386

Clarke L (1997), Zarządzanie zmianą. Gebethner & Ska, Warszawa

Coad P., Yourdon E. (1994), Analiza obiektowa, Oficyna wydawnicza Read me, Warszawa

Cockburn A. (2004), Jak pisać efektywne przypadki użycia, WNT, Warszawa

Copeland D. G, McKenney J. L., (1988), Airline Reservations Systems: Lessons from History, MIS Quarterly, Vol. 11, No. 3, pp. 353–370

Curley K. F., Pyburn P. J., (1982), "Intellectual" technologies: The Key to Improving White-Collar Productivity, Sloan Management Review, Vol. 24, No. 1, pp. 31–39

Czajkiewicz Z. (1998). Klasówka z MRP II. Systemy informatyczne do wspomagania zarządzania produkcją i dystrybucją MRP II/ERP. Computerworld Supplement. June, pp. 62–65

Davenport T. H. (2000) The Future of Enterprise System-Enabled Organizations, Information Systems Frontiers, 2 (2), 163–180

Davenport T. H., Prusak L. (2000), Working Knowledge. Harvard Business School Press, Boston

Davenport T.H. (1998), Putting the Enterprise into the Enterprise System, Harvard Business Review, 76 (4), pp. 121–131

Davenport T.H., Harris J.G., Cantrell S. (2002) The Return of Enterprise Solutions: The Director's Cut, Accenture

Davison R. (2002) Cultural Complications of ERP, Communications of the ACM, 45 (7), 109–111

De Haes S. Van Grembergen, W. V., (2008), An Exploratory Study into the Design of an IT Governance Minimum Baseline through Delphi Research, Communications of AIS, Vol. 22, pp. 443–459

Deloitte Consulting (1998) ERP's second wave: maximizing the value of ERP-enabled processes

DeLone W.H., McLean E.R. (1992) Information Systems Success: The Quest for the Dependent Variable. Information Systems Research 3 (1), 60–95

DeLone W.H., McLean E.R. (2002) Information Systems Success Revisited. Proceedings of the 35th Hawaii International Conference on System Sciences

Doll W. J., Vonderembse M. A., (1987), Forging a Partnership to Achieve Competitive Advantage: The CIM Challenge, MIS Quarterly, Vol. 11, No. 2, pp. 205–220

Doll W.J., Torkzadeh G. (1988) The Measurement of End-user Computing Satisfaction. MIS Quarterly, 12 (2), 259–274

Dos Santos B. L., Peffers K., Mauer D. C., (1993), The Impact of Information Technology Investment Announcements on the Market Value of the Firm, Information Systems Research, Vol. 4, No. 1, pp. 1–23

Durlik I. (1996). Inżynieria zarządzania. Strategia i projektowanie systemów produkcyjnych. Agencja Wydawnicza Placet. Warszawa

Earl M. J., (2003), IT: An Ambiguous Technology?, in: Sundgren B., M?rtensson P., Mähring M., Nilsson K., (Eds.), "Exploring Patterns in Information Management: Concepts and Perspectives for Understanding IT-Related Change", Stockholm School of Economics, The Economic Research Institute, Stockcholm, pp. 39–47

Fitzgerald G. (1998) Evaluating information systems projects: a multidimensional approach, Journal of Information Technology, 13, 15–27

Flakiewicz W., (2002), Systemy informacyjne w zarządzaniu. C.H. Beck, Warszawa

Fogarty D. W., Hoffmann T. R., Stonebraker P. W. (1989). Production and Operations Management. South-Western Publishing Co. Cincinnati

Foster L. W., Flynn D. M., (1984), Management Information Technology: Its Effects on Organizational Form and Function, MIS Quarterly, Vol. 8, No. 4, pp. 229–236

Fowler M, Scott K. (2002), UML w kropelce, Oficyna Wydawnicza LTP, Warszawa

Gable G., Sedera D., Chan T. (2003) Enterprise Systems Success: A Measurement Model. International Conference on Information Systems

Gabryelczyk R. (2006), Aris w modelowaniu procesów biznesu, Difin, Warszawa

Gane C., Sarson T., (1979), Structured Systems Analysis: Tools and Techniques, Prentice-Hall, Engelwood Clifs

Gattiker T., Goodhue D. (2000) Understanding the Plant Level Costs and Benefits of ERP: Will the Ugly Duckling Always Turn Into a Swan, Hawaii International Conference on System Sciences

Goliński, M. (2004). IT sceptycy i IT entuzjaści, w: J. Kisielnicki, J. Nowak, J. Grabara (ed.), „Informatyka we współczesnym zarządzaniu", WNT, Warszawa

Gomółka, Z., (2000), Cybernetyka w zarządzaniu, Agencja Wydawnicza Placet, Warszawa

Grabowski M., (2007), Some Institutional Aspects of Information Technology, Zeszyty Naukowe Wyższej Szkoły Handlowej im. Bolesława Markowskiego w Kielcach, No. 5a, Kielce, pp. 183–192

Grabowski M., (2008a), IT governance – nowy paradygmat zarządzania sferą informatyczną, Zeszyty Naukowe Wyższej Szkoły Handlowej im. Bolesława Markowskiego w Kielcach, No. 8, Vol. 1, Kielce, pp. 431–444

Grabowski M., (2008b), IT governance – zarządzanie sferą informatyczną w ujęciu systemowym, in: Kisielnicki J. (ed.) Informatyka dla przyszłości, Wydawnictwo Naukowe Wydziału Zarządzania Uniwersytetu Warszawskiego, Warszawa, pp.85–93

Grabowski M., (2009), Zrównoważona karta wyników jako narzędzie pomiaru efektywności przedsięwzięć e-biznesowych, Zeszyty Naukowe Uniwersytetu Ekonomicznego w Krakowie, No. 770, pp. 121–134

Grabowski M., (2010), Wybrane zagadnienia ładu informatycznego, Zeszyty Naukowe Uniwersytetu Ekonomicznego w Krakowie, No. 814, pp. 19–33

Grabowski M., Dymek D., (2010), Instytucjonalne aspekty jakości systemów informacyjnych, w: Hales C. F. (ed.), Społeczeństwo informacyjne. Wybrane zagadnienia, Wydawnictwo Uniwersytetu Rzeszowskiego, pp. 139–149

Grabowski M., Zając A., (2009), Dane, informacja, wiedza – próba definicji, Zeszyty Naukowe Uniwersytetu Ekonomicznego w Krakowie, No. 798, pp. 5–23

Greniewski M. J. (1997). MRP II a planowanie strategiczne. [in:] Kubiak, B. F., Korowicki, A. (ed.) Human – Computer Interaction'97. Gdańsk. pp.47–58

Guldentops E. (2004). Governing Information Technology through COBIT, in: W. V. Grembergen (ed.), "Strategies for Information Technology Governance", Idea Group Publishing, Hershey, PA, chapter XI, pp. 269–309

Hammer M., Champy J. (1996) Reengineering w przedsiębiorstwie, Neumann Management Institute, Warszawa

Hammer M., Champy J., (1993), Reengineering The Corporation, Harper Business, New York

Hedman J., Borell A. (2004) Narratives in ERP systems evaluation, The Journal of Enterprise Information Management, 17 (4), 283–290

Holland C., Light B. (1999) A Critical Success Factors Model for ERP Implementation. IEEE Software, May/June, 30–36

Huang S.-M., Chang I.-C., Li S.-H., Lin M.-T., (2004), Assessing risk in ERP projects: identify and prioritize the factors, Industrial Management & Data Systems, 104 (8), 681–688

Huang Z., Palvia P. (2001) ERP implementation issues in advanced and developing countries, Business Process Management Journal, 7 (3), 276–284

Hunton J., Lippincott B., Reck J. (2003) Enterprise Resource Planning Systems: Comparing Firm Performance of Adopters and Nonadopters, International Journal of Accounting Information Systems, 4 (3), 165–184

ITGI (2006), Enterprise Value: Governance of IT Investments, The Val IT Framework; IT Governance Institute, Rolling Meadows, IL

ITGI, (2007), Control Objectives for Information and related Technology (COBIT) 4.1, IT Governance Institute, Rolling Meadows

Ives B., Valacich, J., Watson, R. T., Zmud. R. W., et al.,. (2002). What every Business Student needs to know about Information Systems, Communications of the AIS, Vol. 9, pp. 467–478

Ives B., Vitale M. R., (1988) After the Sale: Leveraging Maintenance with Information Technology, MIS Quarterly, Vol. 12, No. 1, pp. 7–21

Jarvenpaa, S. L., Ives B., (1990), Information Technology and Corporate Strategy: A View from the Top, Information Systems Research, Vol. 1, No. 4, pp.351–376

Johnston H. R., Carrico S. R., (1988), Developing Capabilities to Use Information Strategically, MIS Quarterly, Vol. 12, No. 1, pp. 37–48

Kale V. (2001) SAP R/3. Przewodnik dla menedżerów. Wydawnictwo Helion, Gliwice

Kaplan R., S., Norton D. R., (2001), Strategiczna karta wyników. Jak przełożyć strategię na działanie, Wydawnictwo Naukowe PWN., Warszawa

Kemball - Cook, R. B., (1973), Luka organizacyjna, PWE, Warszawa

Kempisty M. (ed.), (1973), Mały słownik cybernetyczny, Wiedza Powszechna, Warszawa

Kerzner H. (2005) Advanced Project Management. Edycja polska. Helion, Gliwice

Kisielnicki J., (1987), Kryteria jakości systemów informatycznych funkcjonujących w gospodarce narodowej, Wiadomości Statystyczne, No. 10, pp. 6–9

Kisielnicki J., (2008), Systemy informatyczne zarządzania, Wydawnictwo Placet, Warszawa

Kisielnicki J., Sroka H. (2005), Systemy informacyjne biznesu, Wydawnictwo Placet, Warszawa

Klaus H., Rosemann M., Gable G. (2000) What Is ERP? Information Systems Frontiers, 2 (2), 141–162

Koch C. (2001). What is ERP? http://www.darwinmag.com/learn/curve/column.html?ArticleID=39

Kolbusz E., (1993), Analiza potrzeb informacyjnych przedsiębiorstw. Postawy metodologiczne, Uniwersytet Szczeciński, Szczecin

Kotarbiński T., (1975), Traktat o dobrej robocie, Wydanie szóste, Zakład Narodowy im. Ossolińskich – Wydawnictwo, Wocław

Kotarbiński T., (1975), Traktat o dobrej robocie, Wydanie szóste, Zakład Narodowy im. Ossolińskich – Wydawnictwo, Wocław

Kotulski L., Grabowski M., Dymek D., (2007), On the Ability of Supporting IT Audit Demands at the Modelling Level, Proceedings of the 16th International Conference on Computer Science, ISBN 978-83-7493-341-4, Vol.3, pp.326-333 , Wrocław 4–6 September 2007

Koźmiński A. K., Piotrowski W., (1997), Zarządzanie. Teoria i praktyka, Wydanie III, poprawione i rozszerzone, Wydawnictwo Naukowe PWN, Warszawa

Kremers M., van Dissel H. (2000) ERP System Migrations, Communications of the ACM, 43 (4), 53–56

Kroenke, D., (1992), Management Information Systems, McGraw-Hill, New York

Kulik C., Tadeusiewicz R.: Elementy cybernetyki ekonomicznej. Skrypt uczelniany AE, Kraków 1974

Kumar K., van Hillegersberg J. (2000). ERP Experiences and Evolution. Communications of the ACM. 43 (4), 23–26

Kuraś M. (1981), Integracja systemów informatycznych zarządzania, Praca doktorska, Akademia Ekonomiczna w Krakowie

Kuraś M., (1987), Jakość danych a jakość informacji, w: Oleński J. (ed.), "Systemy Informatyczne 1/87", Materiały seminarium SPIS '87, Jakość Danych w Systemach Informacyjnych, Warszawa, pp. 323–331

Kuraś M., (1994), Zmiana organizacyjna jako cel modernizacji systemu informacyjnego, Materiały IV Konferencji Rozwoju Systemów Informatycznych i ich Bazy Sprzętowej w Hutnictwie, Krynica, październik, pp. 93–113

Kuraś M., (2009), System informacyjny – system informatyczny. co poza nazwa różni te dwa obiekty?, Zeszyty naukowe Uniwersytetu Ekonomicznego w Krakowie, No. 770, pp. 259–275

Kuraś M., Grabowski M., Zając A., (1999), Changing IS Curriculum and Methods of Instruction, ACM SIGCSE Bulletin,Vol. 31, No. 3, pp. 36–39

Kuraś M., Zając A. (1999) Czynniki powodzenia i ryzyka projektów informatycznych. Prace z zakresu informatyki i jej zastosowań. Zeszyty Naukowe nr 522. Wydawnictwo Akademii Ekonomicznej. Kraków

Kuraś M., Zając A. (2010), Miejsce, rola i zadania informatyki w zarządzaniu. Ogólny model pojęciowy informatyki w zarządzaniu organizacją. Zeszyty Naukowe Uniwersytetu Ekonomicznego w Krakowie, No. 814, pp. 213–228

Kuraś M., Zając A., (1995), Wykorzystanie metafor organizacyjnych do badania potrzeb użytkowników, VII Górska Szkoła PTI – Szczyrk ,95, Katowice: PTI – Oddział Górnośląski

Langefors B., (1973), Theoretical Analysis of Information Systems, (4th ed.), Studentlitteratur, Lund, Sweden, AUERBACH Publishers Inc., Philadelphia

Laudon K. C., Laudon J. P., (2002), Management Information Systems. Managing the Digital Firm, VII edition, Prentice Hall, Upper Saddle River

Leavitt H. J., Whisler T. J., (1958), Management in 1980's, Harvard Business Review, Vol. 36, No. 6, pp. 41–48

Lech P. (2003) Zintegrowane systemy zarządzania ERP/ERP II. Diffin LST, Warszawa

Light B., Holland C. P., Wills K. (2001) ERP and best of breed: a comparative analysis, Business Process Management Journal, 7 (3), 216–224

Lindsey D., Cheney P. H., Kasper G. M., Ives B., (1990), TELCOT: An Application of Information Technology for Competitive Advantage in the Cotton Industry, MIS Quarterly, Vol. 14, No. 4, pp. 347–357

Loch, L., Venkatraman, N. (1992), Diffusion of technology outsourcing: In?uence sources and the Kodak effect, Information Systems Research, Vol. 3, No. 4, pp. 334–358

Losee R. M. (1997), A discipline independent definition of information, Journal of the American Society for Information Science Vol. 48, No. 3, pp. 254–269. http://ils.unc.edu/~losee/b5/book5.html (retrieved: 3.07.2008)

Lula P., Paliwoda-Pękosz G., Tadeusiewicz R.: Metody sztucznej inteligencji i ich zastosowania w ekonomii i zarządzaniu], Wydawnictwa Akademii Ekonomicznej w Krakowie, 2007

Lyytinen K. (1988) Expectation Failure Concept and Systems Analysts' View of Information Systems Failures: Result of an Exploratory Study. Information Management. No.14

Lyytinen K., Hirschheim R. (1987) Information systems failures – a survey and classification of the empirical literature. Zorkoczy, P.I. (red), Oxford Surveys in Information Technology, Vol. 4, Oxford University Press, Oxford, 257–309

Lyytinen K., King J. L. (2004), Nothing at the center?: Academic legitimacy in the information systems field, Journal of the Association of Information Systems, Vol. 5, No. 6, pp. 220–246

Markus M.L., Axline S., Petrie D., Tanis C. (2000) Learning from adopters' experiences with ERP: problems encountered and success achieved, Journal of Information Technology, 15 (4), 245–266

Markus M.L., Tanis C. (2000), The enterprise system experience – from adoption to success, Zmud, R.W. (ed.), Framing the Domains of IT Management: Projecting the Future Through the Past, Pinnaflex Educational Resources, Inc., Cincinnatti, OH, 173–207

Mazur M. (1967), Cybernetyka, w: Hurwic J. (ed.), „Encyklopedia przyroda i technika", wydanie II, Wiedza Powszechna, Warszawa, p. 227

McNurlin B. C., Sprague Jr. R. H., (2002), Information Systems Management in Practice. Prentice Hall, Upper Saddle River

Meredith J. R. (1992). The management of operations: a conceptual emphasis. John Wiley & Sons, Inc.

Miller G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. Psychological Review, Vol. 63, pp. 81–97, http://www.musanim.com/miller1956/

Moller C. (2005) ERP II a conceptual framework for next-generation enterprise systems? Journal of Enterprise Information Management, 18 (4), 483–497

Muhlemann A.P., Oakland J.S., Lockyer K.G., (1995), Zarządzanie. Produkcja i usługi, Wydawnictwo Naukowe PWN, Warszawa

Murphy K.E., Simon S.J., (2002), Intangible benefits valuation in ERP projects, Information Systems Journal, 12, 301–320

Mynarski S., Szumilak J., Bascik K., Koczyński W., (1989), Elementy teorii systemów i informacji. Akademia Ekonomiczna w Krakowie, Kraków

Nah F., Lau J., Kuang J. (2001) Critical Factors for Successful Implementation of Enterprise Systems. Business Process Management Journal, 7 (3), 285–296

Nah F., Zuckweiler K.M., Lau J. (2003) ERP Implementation: Chief Information Officers Perceptions of Critical Success Factors. International Journal of Human-Computer Interaction, 16 (1), 5–22

Nestor, S. (2001). International efforts to improve corporate governance: Why and how, OECD, http://www.oecd.org/dataoecd/61/1/1932028.pdf

Ngai E. W. T., Law C. C. H., Wat F. K. T. (2008) Examinig the critical success factors in the adoption of enterprise resource planning, Computers in Industry, 59, 548–564

Niedzielska E. (ed.), (2003), Informatyka ekonomiczna, Wydawnictwo Akademii Ekonomicznej we Wrocławiu, Wrocław

Niedźwiedziński M., (1987), Cechy informacji – próba systematyzacji., w: Oleński J. (ed.), „Systemy Informatyczne 1/87", Materiały seminarium SPIS ,87, Jakość Danych w Systemach Informacyjnych, Warszawa, pp. 360–370

Nolan R. L. (1973), Managing the Computer Resource: A Stage Hypothesis, Communications of the ACM, Vol. 16, No. 7, pp. 399–405

Nolan R. L. (1979), Managing the Crises in Data Processing, Harvard Business Review, Vol. 57, No. 2, pp. 115–126

Nolan R. L., Gibson C. F. (1974), Managing the Four Stages of EDP Growth, Harvard Business Review, Vol. 52., No. 1, pp.76–88

North, D. C., (1974), Growth and Welfare in the American Past, 2nd edn. Prentice-Hall, Englewood Cliffs, p.4

Nowicki A., (ed.), (1998), Informatyka dla ekonomistów. Studium teoretyczne i praktyczne, Wydawnictwo Naukowe PWN, Warszawa

Nowicki A., (ed.), (2006), Komputerowe wspomaganie biznesu, Wydawnictwo Placet, Warszawa

Obłój K., (2007), Strategia organizacji. W poszukiwaniu trwałej przewagi konkurencyjnej, Wydanie II zmienione, Polskie Wydawnictwo Ekonomiczne, Warszawa

Ogiela L., Tadeusiewicz R., (2009), Kategoryzacja w systemach kognitywnych, UWND AGH, Kraków

Olson M. H., (1982), New Information Technology and Organizational Culture, MIS Quarterly, Vol. 6, Special Issue, pp. 71–92

O'Shaughnessy, P. (1975), Organizacja zarządzania w przedsiębiorstwie, PWE, Warszawa

Parys T. (1999a). Rozwój systemu zintegrowanego MRP II. Informatyka. No. 5/99. pp.20–27

Parys T. (1999b). MRP II przykładem systemu zintegrowanego. Informatyka. No. 9/99. pp.24–27

Pawlak Z., (1983), Systemy informacyjne. Podstawy teoretyczne

Perreault Y., Vlasic T., (1999), Wdrażając Baan'a IV. Autorski przewodnik prac przygotowawczych i wdrażania, UCL systemy MRP II Spółka Cywilna, Warszawa

Peters, T., Waterman, R. H. (1984), In Search of Excellence. Lessons from America's Best Run Companies. Warner Books, New York

Peterson, R. R. (2004). Information strategies and tactics for information technology governance, in: W. Van Grembergen (ed.), "Strategies for information Technology Governance", Idea Group Publishing, Hershey, PA

Pinto J.K., Prescott J.E. (1988) Variations in Critical Success Factors over the Stages in the Project Life Cycle. Journal of Management, 14 (1), 5–18

Popończyk A. (1996). Dwa w jednym, czyli system informatyczny i system MRP II w przedsiębiorstwie. Informatyka. No. 10/96. pp.11–14

Popończyk A. (1997). Ocena efektywności wdrożenia systemu MRP II. Informatyka Wydanie specjalne MRP II'97. pp.30–33

Primozic K., Primozic E., Laben J., (1991), Strategic Choices: Supremacy, Survival and Sayonara, McGraw-Hill, New York

Rackoff N., Wiseman C. (1985), Information Systems for Competitive Advantage: Implementation of a Planning Process, MIS Quarterly, Vol. 9, No. 4, pp. 285–294

Rainer R. K., Cegielski C. G., (2011), Introduction to Information Systems, JohnWiley & Sons, Inc., Hoboken

Ritchie D. L., (1986), Shannon – and Weaver. Unravelling the paradox of information. Communication Research Vol. 13, No.2, pp. 278–298

Robey D. (1979) User Attitudes and Management Information System Use, Academy of Management Journal 22 (3), 527–538

Rockart J. F. (1979) Chief Executives Define Their Own Information Needs, Harvard Business Review, March, 81–93

Rokicka-Bronitowska A., (ed.), (2004), Wstęp do informatyki gospodarczej, Szkoła Główna Handlowa – Oficyna Wydawnicza, Warszawa

Ross J. W., Vitale M. R. (2000) The ERP Revolution: Surviving vs. Thriving, Information Systems Frontiers, 2 (2), 233–241

Rowley J., (2007), The wisdom hierarchy: representations of the DIKW hierarchy, Journal of Information Science, Vol. 33, No. 2, pp. 163–180

Roztocki N., Weistroffer H.R. (2008) Information Technology in Transition Economies, Jounal of Global Information Technology Management, 11 (4), 1–9

Rutherford, M. (2001), Institutional economics: Then and now, Journal of Economic Perspectives, Vol. 15 No. 3, pp. 173–194

Sambamurthy, V., Zmud, R. V. (1999). Arrangements for information technology governance: A theory of multiple contingencies, MIS Quarterly, Vol. 23, No. 2, pp. 261–298

Sarkis J., Sundarraj R.P., (2000), Factors for strategic evaluation of enterprise information technologies, International Journal of Physical Distribution & Logistics Management, 30 (3/4), 196–220

Scheer A.W., Habermann F. (2000). Making ERP a Success. Communications of the ACM. 43 (4), 57–61.

Schrader A. M. (1983), Toward a Theory of Library and Information Science. (Doctoral dissertation, Indiana University, 1983). Dissertation Abstracts International, AAT 8401534

Scott Morton M. S. (Ed.) (1991), The Corporation of the 1990. Information Technology and Organizational Transformation, Oxford University Press, Oxford

Seddon P.B. (1997) A Respecification and Extension of the DeLone and McLean Model of IS Success, Information Systems Research 8 (3), 240–253

Sedera D., Gable G., Chan T., (2004), A Factor and Structural Equation analysis of the Enterprise Systems Success Measurement Model, Proceedings of the 10th Americas Conference on Information Systems, New York, USA, 676–682

Shang S., Seddon P.B. (2002) Assessing and managing the benefits of enterprise systems: the business manager's perspective, Information Systems Journal, 12, 271–299

Shannon C. E., (1948), A Mathematical Theory of Communication, Bell System Technical Journal, Vol. 27, pp. 379–423; 623–656, http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf

Simon H. (1997), Administrative Behavior: A Study of Decision-Making Processes in Administrative Organizations, The Free Press, New York

Sneller L., Bots J. (2005) Ex-ante Evaluation of ERP Systems: Design and Test of an Evaluation Method for Small and Medium Sized Enterprises, European Conference on Accounting Information Systems

Soja P. (2001). Wdrożenie systemu zintegrowanego klasy MRP II. Zeszyty Naukowe Akademii Ekonomicznej w Krakowie No. 569. pp.85–95

Soja P. (2004). Important Factors in ERP Systems Implementations: Result of the research in Polish enterprises. Proceedings of the 6th International Conference on Enterprise Information Systems ICEIS 2004, Vol.1, Porto, Portugal, pp.84–90

Soja P. (2005) Rozwój zintegrowanych systemów zarządzania klasy ERP. IV Konferencja Entuzjastów Informatyki, Państwowa Wyższa Szkoła Zawodowa w Chełmie, pp.202–208

Soja P. (2006a) Efekty wdrożeń systemów klasy MRP II/ERP. Zeszyty Naukowe Akademii Ekonomicznej w Krakowie No. 724, pp.49–65

Soja P. (2006b) Integracja przedsiębiorstwa jako efekt wdrożenia systemu ERP, „Technologie Informacyjne w Zarządzaniu" Materiały konferencyjne, Wyższa Szkoła Ekonomiczno-Informatyczna, Warszawa, pp.49–58

Soja P. (2006c) Ocena zintegrowanego systemu zarządzania klasy ERP, w: Kisielnicki J. (ed.) Informatyka w globalnym świecie, Wydawnictwo PJWSTK, Warszawa, pp. 334–341

Soja P. (2006d) Success factors in ERP systems implementations: lessons from practice. Journal of Enterprise Information Management, 19 (4), 418–433

Soja P. (2008a) Barriers to Enterprise System Implementation Success: Learning from Practitioners in Poland, Proceedings of the 14th Americas Conference on Information Systems, Toronto, Canada

Soja P. (2008b) Czynniki ryzyka we wdrożeniach zintegrowanych systemów zarządzania przedsiębiorstwem, [in:] Kisielnicki J. (ed.) „Informatyka dla przyszłości" Wydawnictwo Naukowe Wydziału Zarządzania Uniwersytetu Warszawskiego, Warszawa, 202–210

Soja P. (2009) Motivations for Enterprise System Adoption: Insights from Practitioners in Poland. Proceedings of the 15th Americas Conference on Information Systems, San Francisco, California, August 6th–9th 2009.

Soja P. (2010a) Pojęcie sukcesu we wdrożeniu systemu klasy ERP. Prace z zakresu informatyki i jej zastosowań. Zeszyty Naukowe Uniwersytetu Ekonomicznego w Krakowie No. 814. pp.111–127

Soja P., (2010b), Rethinking Critical Success Factors for Enterprise System Adoption: The Case of a Transition Economy, Proceedings of the 16th Americas Conference on Information Systems, Lima, Peru, August 12–15, 1–11

Soja P., Paliwoda–Pękosz G. (2009) What are real problems in enterprise system adoption? Industrial Management & Data Systems, 109 (5), 610–627

Soja P., Stal J. (2008) Uwarunkowania międzyorganizacyjnej integracji systemów wspomagających zarządzanie przedsiębiorstwem, Lachiewicz S., Zakrzewska-Bielawska A. (ed.) Zarządzanie przedsiębiorstwem w warunkach rozwoju wysokich technologii, Monografie Politechniki Łódzkiej, Wydawnictwo Politechniki Łódzkiej, 269–278

Somers T., Nelson K. (2004) A taxonomy of players and activities across the ERP project life cycle. Information & Management, 41, 257–78

Somers T., Nelson K., Karimi J. (2003) Confirmatory Factor Analysis of the End-User Computing Satisfaction Instrument: Replication within an ERP Domain, Decision Sciences, 34 (3), 595–621

Stabryła A., (1995), Podstawy zarządzania firmą, Wydawnictwo Naukowe PWN, Warszawa

Stamper R. (1973), Information in business and administrative systems, Willey, New York

Stefanou C.J. (2001) A framework for the ex-ante evaluation of ERP software, European Journal of Information Systems, 10, 204–215

Stefanowicz B., (1987), Jakość informacji w ujęciu infologicznym. Wiadomości statystyczne, No. 1, pp. 29–31

Stefanowicz B., (2007), Informacyjne systemy zarządzania. Przewodnik, Szkoła Główna Handlowa w Warszawie, Warszawa

Steinműller W. (1977), Zautomatyzowane systemy informacyjne w administracji prywatnej i publicznej, Organizacja Metody Techniki, 1977/9

Stewart, T. A., Brown, J. S., Hagel III, J., McFarlan, W. F., Nolan, R. R., Carr, N. G. (2003), Does IT matter? an HBR debate, Harvard Business Review, Vol. 81, No. 7, pp. 109–112

Stoner J. Ch. Wankel, (1992), Kierowanie, Państwowe Wydawnictwo Ekonomiczne, Warszawa

Sumner M. (1999) Critical success factors in enterprise wide information management systems projects. Proceedings of 5th Americas Conference on Information Systems, 232–234

Sumner M. (2000) Risk factors in enterprise-wide-ERP projects, Journal of Information Technology, 15, 317–327

Sundgren B., Steneskog G, (2003), Information Systems for Concerted Actions, in: Sundgren B., Mårtensson P., Mähring M., Nilsson K., (Eds.), “Exploring Patterns in Information Management: Concepts and Perspectives for Understanding IT-Related Change”, Stockholm School of Economics, The Economic Research Institute, Stockcholm, pp. 11–38

Sundgren, B. (1973), An Infological approach to Data Bases. Doctoral thesis, University of Stockholm, Department of Administrative Information Processing. Central Bureau of Statistics, Sweden

Sutherland E., Morieux Y., (1988), Effectiveness and Competition-Linking Business Strategy, Organizational Culture and the Use of Information Technology, Journal of Information Technology, Vol. 3, No. 1, pp.43–47

Szyjewski, Z., (2004). Metody zarządzania projektami informatycznymi. Agencja Wydawnicza „Placet", Warszawa

Szymanik J., Zajenkowski H. (2004), Kognitywistyka – o umyśle umyślnie i nieumyślnie. Uniw. Warsz., Warszawa

Tadeusiewicz R. (1994). Problemy biocybernetyki, Wydawnictwo Naukowe PWN, Warszawa

Tadeusiewicz R., Ogiela L.(2008), Selected Cognitive Categorization Systems, Chapter in book: Rutkowski L. et al. (Eds.): Artificial Intelligence and Soft Computing, ICAISC 2008, Lecture Notes on Artificial Intelligence, vol. 5097, Springer-Verlag, Berlin-Heidelberg, pp. 1127–1136

Tadeusiewicz R., Ogiela M. R. (2004), Medical Image Understanding Technology, Series: Studies in Fuzziness and Soft Computing, Vol. 156, Springer-Verlag, Berlin – Heidelberg – New York

Tavakolian, H. (1989), Linking the Information Technology Structure with Organizational Competitive Strategy: A Survey, MIS Quarterly, Vol. 13, No. 3, pp. 309–317

Teltumbde A., (2000), A framework for evaluating ERP projects, International Journal of Production Research, 38 (17), 4507–4520

The Office of Government Commerce (2009), Managing and Directing Successful Projects with PRINCE2®, http://www.best-management-practice.com/gempdf/PRINCE2_2009_Overview_Brochure_April2011.pdf, dostęp 21 kwietnia 2011

The Scrum Guide (2011) http://www.scrum.org/storage/scrumguides/Scrum%20Guide.pdf, dostęp 21 kwietnia 2011

Trzupek M., Ogiela M. R., Tadeusiewicz R. (2009), Medical pattern intelligent recognition based on linguistic modelling of 3D coronary vessels visualizations. Elektronika No. 11, 2009, pp. 9 – 12

Umble E.J., Umble M.M., (2002), Avoiding ERP implementation failure, Industrial Management, January/February, 25–33

Unold J., (2004), System informacyjny a jakościowe ujęcie informacji, w: Porębska-Miąc T., Sroka H., (ed.), "Systemy Wspomagania Organizacji SWO'2004", Wydawnictwo Akademii Ekonomicznej im. Karola Adamieckiego w Katowicach, Katowice, pp. 163–170

Van Grembergen W., R. Saull, S. De Haes, (2004): Linking the IT Balanced Scorecard to the Business Objectives at a Major Canadian Financial Group, W. Wan Grembergen, "Strategies for Information Technology Governance", Idea Group Publishing, London, (pp. 129–151)

Van Grembergen, W. (2002). Inroduction to the minitrack: It governance and its mechanisms, Proceedings of the 35th Hawaii International Conference on System Sciences (HICCS)

Van Grembergen, W., de Haes, S. (2004). Structures, processes and relational mechanisms for IT governance, in: W. V. Grembergen (Ed.), "Strategies for Information Technology Governance", Idea Group Publishing, Hershey, PA, chapter I, pp. 1–36

Velcu O. (2007) Exploring the effects of ERP systems on organizational performance. Evidence from Finnish companies, Industrial Management & Data Systems, 107 (9), 1316–1334

Volkoff O., Strong D.M., Elmes M., (2005), Understanding enterprise systems-enabled integration, European Journal of Information Systems, 14, 110–120

Wallace T. F. (1990). MRP II: Making It Happen. The Implementers' Guide to Success with Manufacturing Resource Planning. Oliver Wight Publications Inc. Essex Junction

Ward, J. M. (1988), Information Planning for Strategic Advantage, Journal of Information Technology, Vol. 3, No. 3, pp. 169–177

Weaver W., (1949), Recent Contributions to the Mathematical Theory of Communication, in. Shannon C.E. Weaver W. (1949), "The Mathematical Theory of Communication", University of Illinois Press, Urbana, http://academic.evergreen.edu/a/arunc/compmusic/weaver/weaver.pdf

Weill P., (1992), The Relationship Between Investment in Information Technology and Firm Performance: A Study of the Valve Manufacturing Sector, Information Systems Research, Vol. 3, No. 4, pp. 307–333

Weill P., Broadbent, M. (1998). Leveraging the New Infrastructure: How Market Leaders Capitalize on IT, Harvard Business School Press, Boston, MA

Weill P., Ross, J. W. (2004). IT Governance, Harvard Business School Press, Boston, MA

Weill P., Subramani M., Broadbent M., (2002), IT Infrastructure for Strategic Agility, Center for Information Systems Research, MIT, WP No. 329

Welke R.J., Konsynski B.R. (1980) An Examination of the Interaction Between Technology, Methodology and Information Systems: A Tripartite View, Proceedings of the First International Conference on Information Systems, 32–48

Wiener N., (1954). Cybernetics in History. In The human use of human beings: Cybernetics and society, pp. 15–27, Houghton Mifflin, Boston.

Wrycza S, Marcinkowski B., Wyrzykowski K. (2005), Język UML 2.0 w modelowaniu systemów informatycznych, Helion, Gliwice

Wrycza S. (1999), Analiza i projektowanie systemów informatycznych zarządzania, Wydawnictwo Naukowe PWN, Warszawa

Wrycza S. (ed.), (2000), Informatyka dla ekonomistów, Wydawnictwo Uniwersytetu Gdańskiego, Gdańsk

Wrycza S. (ed.), (2010), Informatyka ekonomiczna, PWE, Warszawa

Wysocki R.K., McGary R. (2005) Efektywne zarządzanie projektami. Wydanie III. Helion, Gliwice

Yourdon E., (1996), Współczesna analiza strukturalna, WNT, Warszawa

Zając, A., (2004), Metafory organizacyjne w badaniu potrzeb informacyjnych przedsiębiorstwa, Praca doktorska, Akademia Ekonomiczna w Krakowie

Zawiła-Niedźwiecki J., Rostek K., Gąsiorkiewicz A., (ed.), (2010), Informatyka gospodarcza, Tomy 1–4, Wydawnictwo C.H. Beck, Warszawa

Zeleny M., (1987), Management Support Systems: Towards Integrated Knowledge Management, Human Systems Management, Vol. 7, No. 1, pp. 59–70

Zieleniewski J., (1981), Organizacja i zarządzanie, Wydanie VII, Państwowe Wydawnictwo Naukowe, Warszawa

Zmud R., W., (1984), Design Alternatives for Organizing Information Systems Activities, MIS Quarterly, Vol. 8, No. 2, pp. 79–93

Zuboff S. (1985) Automate/informate: the two faces of intelligent technology, Organizational Dynamics, 14 (2), 5–18