Tilburg University

DSS, information systems and management games

Casimir, R.J.

Publication date:
1986

[Link to publication in Tilburg University Research Portal](#)

Citation for published version (APA):

Casimir, R. J. (1986). *DSS, information systems and management games*. (Ter discussie FEW; Vol. 86.07). Unknown Publisher.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

CBM

R

7627

1986

7

* C I N O O 2 2 7 *

faculteit der economische wetenschappen

REEKS "TER DISCUSSIE"

Katholieke
Hogeschool
Tilburg

No. 86.07

DSS, Information systems and Management

Games

652.45

Robert J. Casimir

mei 1986

DSS, Information systems and Management Games

Rommert J. Casimir

1 Introduction

A Decision Support System (DSS) is usually defined as an information system that supports unstructured or semistructured decisions [1]. Because DSS's often use sophisticated modeling techniques and specialised programming languages and environments, many authors stress the importance of "DSS generators" and "DSS tools" to the point that the techniques used in the design of an information system rather than its purpose are considered the main characteristic of a DSS. So a DSS may be considered successful even if it was ultimately used for quite other purposes than management support. This has led Huber [11] to make a distinction between "DSS", which he broadly catagorizes as the products that are advertized under that name and "dss" or "decision support systems", which he defines as the information systems that are actually used by managers for decision support.

In this paper I will use the term Decision Support System for an information system that provides support for managagerial decisions and I will try to sharpen this definition by considering the different types of decisions in an organization. In section 3 this classification will be applied to the decisions in a management game, which of course is a model of an organization. Section 4 contains a description of INFOLAB, a management game we are designing with the express aim of using it as a tool in Information System research and education.

2 Decisions and Information Systems

We can distinguish two main types of information systems in organizations:

- a) Systems that define the rights and obligations of an organization or of groups or individuals within the organization.
- b) Systems that provide information for decisions at any level.

Information systems of the first type provide the bulk of manual and electronic data processing. Examples are accounting for accounts receivable, salary systems, and sales and purchase accounting for VAT. DSS texts often criticize this type of information system for not providing adequate information for management, thereby neglecting its important role. Not all transaction systems belong to this class, because many transaction systems incorporate decisions at operational levels, e.g. an order entry system may decide how an order for an out-of-stock item has to be handled. Management control systems and operational control systems constitute an important subclass of the first type, because their primary aim is to ascertain whether instructions have been carried out correctly. In section 4 we will consider the use of INFOLAB for research in management control systems.

Information systems of the second type are called management information systems or management accounting systems. This class of information systems includes, but is not equivalent to Decision Support Systems. If an information system serves both ends we say that the organization has two identical information systems.

2.1 Classification of decisions

Two important criteria for the classification of decisions are the relative importance of the decision and the time that is available for the decision. A decision is important if the organization risks a large part of its resources by the decision. Thus a decision of a doctor or a tax inspector is unimportant or minor for the decision maker although it may be a question of life and death (or rags or riches) for the client. The importance of a decision is the product of the relative risk and the amount involved. So the decision on a \$100,000 loan to a small businessman may be as important as the decision on a one billion dollar loan to the US government. Risk is equivalent to uncertainty, which is also a yardstick for the amount of information that is needed for the decision. Thus for decisions involving the same amount of money, the importance of a decision is directly proportional to the amount of information needed. The notion of the importance of a decision presupposes that the risk and the amount involved can be estimated beforehand. If we cannot supply this estimate, all decisions must be deemed equally important, so a classification along this axis is impossible.

The importance of a decision is related to the distinction between structured and unstructured, or programmed and nonprogrammed decisions as defined by Simon [21]. Because programmed decisions are defined as being routine, an important decision can never be programmed. Nevertheless, the notion that an important decision can be programmed is sometimes found in DSS or operation research literature. This is because an individual decision may be replicated in a large number of individuals, and as a consequence a generally accepted rule may be established. A decision using such a rule may be called a socially programmed decision. On the other hand, not all unimportant decisions can be programmed either. The work of professionals, like doctors, lawyers and teachers, mainly consists of taking minor nonprogrammed decisions. Such decisions are generally neglected in organizational literature.

The length of the time available for a decision is considered long or short in relation to the time needed to gather and process the information. Time is short if dangers lie ahead or if opportunities may vanish. This occurs notably if there is a risk of idleness of expensive resources. Time is long if not much is lost by postponing a decision.

For each type of decision the decision maker needs an appropriate type of information system. Decisions and information systems are listed in table 1.

Table 1: Decisions and Information Systems

	short	long
unimportant	operational decisions	bureaucratic decision
	operational system	bureaucratic system
important	crisis decision	strategic decision
	?	decision support system

Operational decisions

Operational decisions are defined by Anthony [2] as decisions that are normally taken by operating (in contrast to managerial) personnel. However, this definition cannot be applied in organizations where employees regularly combine operating and managerial tasks. By defining operational decisions as unimportant decisions that have to be taken at short notice we make no supposition about the decision makers. Because operating decisions have to be made at short notice, they leave no time for consultation with managers. Accordingly, they have to be made by a computer, by operating personnel following predefined rules, or, certainly if they are of the nonstructured variant, by a professional. Examples of operational decisions are ordertaking and buying.

Bureaucratic decisions

With bureaucratic decisions, recourse to a higher-placed manager is always possible. Examples of bureaucratic decisions are tax assessment, graduation and salary review. There is a tendency to use exact rules in bureaucratic decisions because such decisions may be legally challenged. A challenge is possible because persons affected by the decision usually have full information, which is not the case with operational decisions.

Crisis decisions

The distinguishing feature of crisis decisions is that the decision maker lacks the time to collect and process the necessary information. Society as a whole as well as individual organizations try to prevent the need for crisis decisions. An example dating back to prehistoric times is the replacement of hunting by farming. Profits from a speedy reaction to opportunities is sometimes considered immoral, as may be concluded from the fact that use of inside information is illegal in some situations. Accordingly, crisis decisions mainly occur after disasters and in international politics.

Strategic decisions

For strategic decisions there is ample time to collect and process the necessary information. That does not imply that the decision maker has perfect information, because some information may be unavailable or too expensive. An example of a strategic decision in a business context is the investment decision.

Information Systems

Every type of decision asks for an appropriate information system. First I have to remark that it is impossible to design an information system for crisis decisions, because crisis decisions are characterized by the fact that it is impossible to know beforehand what information is needed. Accordingly Parnas [19] and Lin [16] have pointed to the unfeasibility of an automated defense system as proposed by the Reagan administration under the SDI heading. Consequently I further omit information systems for crisis decisions. Table 2 lists the characteristics of the remaining information systems. Although I use computer science terminology in this table, the concepts are valid also if the information systems are implemented with other means.

Table 2: Characteristics of information systems

Information system	System	Technical
	Characteristics	Characteristics
Operational	Simple	Real-time
information system	algorithms	Reliability
Bureaucratic	OR (e.g. LP)	Batch
informatiesysteem	Statistical	Large data sets
	techniques	
DSS	Models	Flexible programs
		Data bases

Transfer of techniques

The techniques that are used in one type of information system can be transferred to another type of information system. So it is not possible to determine the type of information system from the techniques used. To give an example, a model that has been developed to support a top banker in million dollar loan decisions certainly is part of a DSS. However, if the same model is used to decide monthly on a large number of loan applications, it is used in a bureaucratic system. If the model is used by bank clerks to instantly decide on loan applications it is part of an operational system. Accordingly, a DSS may serve as a prototype for a bureaucratic or an operational information system. The similarity of the techniques used in the design of DSS and those used in prototyping is not accidental. The tendency for DSS to evolve to operational systems is called migration by Moore and Chang [18].

3 Management Games

3.1 Simulation of systems

In systems we can distinguish an operational system and a decision system which decides what actions the operational system has to take. The decision system either reacts to a set of specified rules or to a more general program. Decisions which are made according to specific rules are called programmed in the terminology of Simon [21], other decisions are called nonprogrammed. To take decisions in an operational system information (often called feedback) is needed. We assume that this is supplied by a separate information system. The information system extracts this information from the operational system. Most simulation studies center on the decision system part of systems. They take for granted that the information system will extract all necessary information from the operational system and pass it to the decision system. The work of Kleijnen [14], who studied the influence of incomplete information on decisions, is an exception.

3.2 Traditional management games

A management game is a game where two or more players have to take managerial decisions in a simulated world. A popular type of management game is the business game where the simulated world contains a number of competitive business units. In a business game participants have to take decisions on price, marketing expenses, production and capital expenditure. A description of a large number of management games is given by Elgood [10]. The level of abstraction in a management game is far lower than the level of abstraction in game theory. Management games are mainly used in education. Many researchers prefer to use interactive simulations, also named simulation games. In those games a single player tries to maximize his results in a simulated world.

A traditional management game is played in a discrete number of rounds or periods, where each round is equivalent to a month, a quarter or a year in the simulated world. The decisions for a round are taken simultaneously by all players. Consequently players never have complete information on the state of play. When all players have entered their decisions the results of a round are computed and reported to the players. The time available for decision making by the players normally is much longer than the time needed to compute the results. An average proportion is half an hour of decision time against five minutes of computing time. In traditional management games data are collected and processed according to a standard method, with the exception of marketing data that normally have to be bought.

Management games and information systems

To apply the classification of decisions given in section 1 to management games, we replace the terms used in that table by terms derived from management games. This translation is given in table 3.

Table 3: Translation of terms

Term from table 1	Equivalent in management game	
Short	During a round	
Long	Between two rounds	
Unimportant	Simulated by computer	
Important	Decided by player	

From this translation we compose table 4, which specifies the information systems occurring in a management game. It should be noted that crisis decisions cannot be made in a game that is played in rounds, because players cannot interfere with the simulated world during a round. A game where players can interfere at any moment should be designed as a real-time system and it should organise its output in a way that encourages fast reaction. Experience with arcade games [6] learns that graphical output is preferable for that purpose. Technically, a real-time game may be approximated by shortening the time available for decisions and diminishing the number of decisions that is required in each round. We should also note that minor, unstructured decisions, which probably occupy the real-life manager during a large part of his working life, are typically absent from management games. This has no consequence for their educational value, because students have ample opportunity to practise such decisions in other contexts.

Table 4: Information systems in a management game

	During a round	Between rounds	
By computer	Operational	Preprogrammed decisions	
	program		
By player		DSS for player	

4 INFOLAB

INFOLAB, a laboratory for information systems, is a management game that can be used as an environment for the development and use of information systems. The project was initiated by Kleijnen [15], a preliminary description was given by Casimir [4]. Just as in a conventional management game, players (either students or managers) take management decisions for simulated organizations in a competitive environment. In the INFOLAB prototype these organizations will be modeled after industrial companies, but in future versions the type of organization may be defined to a large extent by the game administrator.

The main difference between INFOLAB and conventional management games is in the output produced by the game for the players. Conventional management games produce standard reports in a format normally used by management, i.e. financial statements and statistical summaries. In INFOLAB the player has to specify what types of events should be registered. The output consists of the raw data on those events. The player is indeed free to do no accounting at all and base his decisions on his cash level (which the bank will report free) and market research reports. The player who wants to use other data for decision making will have to build an information system to process the raw data, because the amount of data provided to any player will be too large for manual handling (say 1000 items).

4.1 INFOLAB versions

In accordance with table 4 we distinguish two versions of INFOLAB:

INFOLAB-1 The information system in the game supplies data that can be used by players at the end of each round.

INFOLAB-2 The information system in the game also supplies data to a decision system that is built into the game.

So INFOLAB-1 supports Decision Support Systems and bureaucratic information systems as well as control systems. INFOLAB-2 also supports operational information systems. In INFOLAB-1 information processing is divided into two distinct phases: data collection and data

reduction. The first phase is executed during a round. Specification of this part of the information system consists of pinpointing the variables that have to be measured.

The second phase is executed between rounds. This part of the information system is defined by user-written program. For these programs any convenient language or program package, including spreadsheets and database systems, may be used.

In INFOLAB-2 the data collected by the information system has to be analyzed during a round to signal actions to the decision system. To this end a special language will be designed that allows players to define information system procedures and decision system procedures inside the simulated environment. The design of this language poses a number of problems which will be detailed in section 4.3. However, development of INFOLAB-2 is important because, contrary to the opinion of many DSS authors, building operational information systems is the central and most difficult information systems activity. Because of the difficulty of implementation of INFOLAB-2, we have decided to start with the development of INFOLAB-1 and only start the development of INFOLAB-2 when the mechanisms of the game and player attitudes have been studied.

4.2 INFOLAB-1

Outline of the game

The gaming model of INFOLAB is derived from MAGEUR, a traditional management game I developed for Erasmus University [3]. The main entity in the game is the firm, which is managed by the participant or player. Each firm produces a number of products, which may be chosen from opportunities, which in turn are supplied by research. A product is produced by a product line which employs machines, workers and factory space. Production also entails the use of materials. Firms may switch production factors among different products, discontinue products and introduce new products as they see fit. Sales are made to simulated consumers who choose among competitors on a number of determinants, like price, quality and marketing effort. Typically, a firm produces some tens of product batches and effects some hundreds of separate sales transactions for each

product. All production and sales results are reported separately as accounting and statistical computation are the task of the participants. However, the game does some operational accounting, e.g. it will compute the cash level to use it as a base for interest charges and it will compute the stock level of materials and finished products to establish production and sales opportunities.

Most of the rules governing the operation of the game will be built into the game program, e.g. sales are possible only if a product is in stock, but players may be given some scope for the definition of rules, e.g. a reorder level and an order quantity may be specified for materials.

The game program will be divided into two parts: INFOLAB will provide the interface with the participants. INFOMARK will compute the results for a round. Supplementary programs BANKER and STOCK will establish the interface with the banker and the stock exchange. The division of the system into separate programs is sound from a logical point of view. It is also necessary because the prototype will be implemented in an environment with limited program space (i.e. Turbo Pascal).

Apart from the usual decisions on production, marketing and finance the participant in INFOLAB also can define two types of data to be collected: data on events and data on states. Collection of event data takes place at any occurrence of the stated event, collection of state data (inventory taking) occurs at specified timing intervals. In information system design it is well known that data cannot be collected free. So there is a price attached to all data collection. Because INFOLAB is a discrete event simulation, the number of events, and consequently the number of different events, is finite, so there is a finite number of possible measurements. The set of actual measurements will be defined by the player by explicit choice from a given set. To choose an appropriate set of measurements, the player will have to thoroughly study the model of the firm. Specifically, he will not be allowed to opt for the full range of outputs, because this will be far too expensive.

INFOLAB-1 for decision support

When INFOLAB-1 is used for decision support the participant will get ample time (say a week after each round) to study the results of the last round. The choice of a programming tool to investigate the data file is completely left to the participant. However, the datafile will be too large for manual handling. Exploratory research will try to establish some relation between tools and models used and results in subsequent periods. In this respect INFOLAB is free of prejudice. It is quite possible that it will prove the superiority of conventional financial reporting systems over Decision Support Systems as a base for managerial decisions.

In a later phase we want students in Information Systems to cooperate with students in Marketing, Finance and Accounting, with the former as information system builders and the latter as users. In this phase the user will get less time for his analysis (say one hour for a round), but the builder will get considerable more than a week's time before the start of the first round. This use of INFOLAB-1 has some likeness to the so-called Minnesota experiments [7] and subsequent research [5,12,13,8]. The aim of this research has been to investigate the relation between decision quality and type of DSS. To this end researchers applied preprogrammed DSS's that used the standard output of a conventional business game as data, whereas research with INFOLAB will primarily try to learn whether decision makers can judge the quality of a DSS and whether DSS builders know to find the appropriate data.

INFOLAB-1 for bureaucratic decisions

So far we have assumed that players in INFOLAB-1 will only take strategic decisions. However, there is no logical objection to introducing bureaucratic decisions into INFOLAB, e.g. by enlarging the number of products, the number of production factors involved in the production of any product, or the number of research proposals that have to be studied before a possibly successful product is found. Bureaucratic decisions for the next round, e.g. pricing decisions for a large number of products may be taken by programs using the data of the latest round. This follows research by Shubik et al. [20], Kleijnen [14] and Dickinson and Ferrel [9], who all studied robot players in management games.

INFOLAB-1 for control systems

The principal aim of management control systems is to establish from data on operations whether management instructions have been carried out correctly. The introduction of a management control system into INFOLAB makes sense only if actual operations may differ from the instructions issued by the player and if the player can influence those differences. To this end we introduce daemons into the game. A daemon is a random mechanism that changes some specific variable, e.g. a cash daemon reduces the cash level at unpredictable moments. The activity of a daemon can be stopped, but only if it has been discovered. This is implemented by increasing the cost of stopping a daemon with decreasing activity. Thus, prevention of daemon activity will have to start with getting a correct impression of its level from a management control system. The notion of a daemon is universal. Daemon activity encompasses fraud and other crimes, human errors, machine failures, epidemics, strikes and acts of God. Daemons may be defined for any object, including accounting records. If daemons for accounting records are included, outputs of the management control system should be accepted only after careful auditing.

An example

If there is no daemon activity in the production area, the amount of stock in hand can be computed from the formula:

$$\text{present stock} = \text{previous stock} + \text{production} - \text{sales}$$

However, if a stock daemon is active, we have to compute the loss of stock with the formulas:

$$\text{present stock} = \text{inventory}$$

$$\text{loss of stock} = \text{previous stock} + \text{production} - \text{sales} - \text{present stock}$$

If we also assume the presence of a production record daemon, we will have to independently ascertain the production level by comparing production with input of production factors, by more frequent inventory taking (because we may assume that the chance of simultaneous activity of both daemons is smaller then) or by replicating production reporting through a different channel.

Technically, there is no difference between information system building for decision support and information system building for management control, as the same tools may be used. However, the user groups are quite different, as the use of INFOLAB-1 for management control will be primarily intended for EDP audit courses.

4.3 INFOLAB-2

In INFOLAB-2 students in Information Systems will have to design and build an information system and a decision system for a simulated firm in the game. As was stated before, the central design problem of INFOLAB-2 is the design of a language that may be used to define such systems. That language should be easy to use, but it should also be complete in the respect that it should allow the definition of business policies. In the design of the language I prefer to start from a well defined concept and proceed by deleting unnecessary features and adding needed features. "Fourth generation languages" are not mentioned because the concept is not well defined. I propose to study the following six programming models:

1 Collections of standard programs

Instead of designing an information system from scratch, many users build it from predefined packages. Use of this approach has two advantages: Implementation is straightforward and the design process for users can be made as easy as necessary. However, there are also two objections against this model. First, writing application programs or porting standard packages into INFOLAB is not an attractive job and second, information systems builders are often considered system architects and it would be rather inappropriate to limit the education of an architect to a course in choosing the right prefabricated building.

2 Procedural languages

At the other extreme we may use a procedural language. A procedural language is certainly powerful enough for the job. Also, students in Information Systems usually have a fair experience with programming in a procedural language. The compilation or interpretation of a

procedural language poses no problem as it is a well-studied subject in Computer Science. A procedureal language can be derived from Pascal by simplifying data and control statements. The advantage of standard programs may be introduced by adding preprogrammed procedures. The main disadvantage of using a procedural language is that programming problems may tend to dominate system design problems.

3 Nonprocedural languages

A nonprocedural language only specifies what has to be computed, and not how it has to be computed. Consequently, execution of programs in a nonprocedural language can be intolerably inefficient, e.g. when sorting a list is described as finding a permutation of that list that happens to be sorted. Although nonprocedural languages are considered userfriendly, their novelty, as well as some special characteristics, like heavy reliance on recursion as in Prolog, may deter Information System students. Moreover, decision systems are sometimes better defined in a procedural language, because actual business policy is often described in a procedural fashion.

4 Query languages

Query languages offer powerful operations on relations. However, they normally lack mechanisms for recursion or iteration. A query language without recursion or iteration may be too restricted for decision system definition, a query language with either of those mechanisms is in fact a procedural or nonprocedural language with an improved datastructure.

5 Spreadsheets

A spreadsheet is a program in a special type of nonprocedural language. In contrast to query languages, spreadsheets define operations on single cells, but facilitate definitions by copying. Iteration is provided by repeated evaluation, and is restricted only by the size of the spreadsheet. Spreadsheets are effective tools for simple computations, but their potential for more elaborate applications has not been thoroughly explored.

6 Decision tables

Decision tables have proved their value for the definition of decisions [17]. Moreover, both the use and implementation of decision tables are very simple. Consequently, decision table may be introduced into INFOLAB-1 as a first step toward INFOLAB-2. However, because they lack iteration and recursion, the power of decision tables is limited.

From this discussion it should be clear that some experimental studies in information system design should be made before the implementation of INFOLAB-2 is started. INFOLAB-1 will provide a suitable environment for those experiments, especially when programs for bureaucratic decisions are being written.

5 Conclusion

Management games can be used to better define the distinction between different types of information systems and their use in organizations. Consequently, research in the problems of informations systems development in management games will be valuable for a wide range of real-life applications.

References

- 1 Alter, S.L. Decision support systems: current practices and continuing challenges. Addison-Wesley, Reading Mass., 1980.
- 2 Anthony, R.N. and J.S. Reece: Accounting text and cases. Irwin, Homewood Ill., 1979.
- 3 Casimir, R.J. Mageur, Handleiding ondernemers. (Mageur, participants manual) Erasmus University, Rotterdam, 1985.
- 4 Casimir, R.J. INFOLAB, een laboratorium voor informatiesystemen. (INFOLAB, a laboratory for information systems) Reeks "ter discussie" 85.14 (working paper), Tilburg University, Tilburg 1985.

- 5 Courtney, J.F., G. DeSanctis and G.M. Kasper: Continuity in MIS/DSS laboratory research: the case for a common gaming simulator.
Decision sciences Vol 14 No 3 (july 1983), pp 419-439.
- 6 DeFanti, T.A.: The mass impact of videogame technology.
in: Advances in Computers, Vol 24 (edited by M.C. Yovits), Academic Press, Orlando, 1984.
- 7 Dickson, G.W., J.A. Senn and N.L. Chervany: Research in management information systems: the Minnesota experiments.
Management science Vol 23 No 9 (may 1977) pp 913-923.
- 8 Dickson, G.W., G. DeSanctis and D.J. McBride: Understanding the effectiveness of computer graphics for decision support: a cumulative experimental approach.
CACM Vol 29 No 1 (jan 1986), pp 40-47.
- 9 Dickinson, D and W.R. Ferrell: Fuzzy set knowledge representation in a system to recommend management decisions.
in: Methlie, L.B. and R.H. Sprague (editors):
Knowledge representation for decision support systems.
North Holland, Amsterdam, 1985.
- 10 Elgood, C.: Handbook of management games 2nd ed.
Gower, Aldershot, 1981.
- 11 Huber, G.P.: Organization science contribution to the design of decision support systems. in Fick, G. and R.H. Sprague (editors): Decision support systems: issues and challenges.
Pergamon press, Oxford, 1980.
- 12 Jarvenpaa, S.L., G.W. Dickson and G. DeSanctis: Methodological issues in experimental IS research: experiences and recommendations. MIS quarterly Vol 9 No 2 (june 1985).
- 13 Kasper, G.M. and R.P. Cervany: A laboratory study of user characteristics and decision-making performance in end-user computing. Information and management 9 (1985) pp 87-96.

- 14 Kleijnen, J.C.P.: Computers and profits.
Addison-Wesley, Reading Mass., 1980.
- 15 Kleijnen, J.C.P.: De rol van wiskundige modellen en technieken in de bestuurlijke informatica. (The role of mathematical models and techniques in information systems)
Informatie 23 Nr 6 (juni 1981), pp 387-393
- 16 Lin, H. The development of software for ballistic-missile defense.
Scientific American, Vol 253 No 6 (dec 1985), pp 32-39.
- 17 Maes, R., J. Vanthienen and M. Verhelst:
Practical experiences with the procedural decision modeling system. in Sol, H.G. (editor): Processes and tools for decision support. North Holland, Amsterdam, 1983.
- 18 Moore, J.H. and M.G. Chang: Meta design considerations in building DSS. in Bennett, J.L. (editor): Building decision support systems. Addison-Wesley, Reading Mass., 1983.
- 19 Parnas, D.L. Software aspects of strategic defense systems.
CACM Vol 28 No 12 (Dec 1985), pp 1326-1335.
- 20 Shubik, M., G. Wolf and S. Lockhart: An artificial player for a business market game.
Simulation and games Vol 2 Nr 1 (march 1971), pp 27-43.
- 21 Simon, H.A.: The new science of management decision.
Prentice Hall, Englewood Cliffs N.J., 1977.

- 14 Kleijnen, J.C.P.: Computers and profits.
Addison-Wesley, Reading Mass., 1980.
- 15 Kleijnen, J.C.P.: De rol van wiskundige modellen en technieken in de bestuurlijke informatica. (The role of mathematical models and techniques in information systems)
Informatie 23 Nr 6 (juni 1981), pp 387-393
- 16 Lin, H. The development of software for ballistic-missile defense.
Scientific American, Vol 253 No 6 (dec 1985), pp 32-39.
- 17 Maes, R., J. Vanthienen and M. Verhelst:
Practical experiences with the procedural decision modeling system. in Sol, H.G. (editor): Processes and tools for decision support. North Holland, Amsterdam, 1983.
- 18 Moore, J.H. and M.G. Chang: Meta design considerations in building DSS. in Bennett, J.L. (editor): Building decision support systems. Addison-Wesley, Reading Mass., 1983.
- 19 Parnas, D.L. Software aspects of strategic defense systems.
CACM Vol 28 No 12 (Dec 1985), pp 1326-1335.
- 20 Shubik, M., G. Wolf and S. Lockhart: An artificial player for a business market game.
Simulation and games Vol 2 Nr 1 (march 1971), pp 27-43.
- 21 Simon, H.A.: The new science of management decision.
Prentice Hall, Englewood Cliffs N.J., 1977.

IN 1985 REEDS VERSCHENEN

01. H. Roes	Betalingsproblemen van niet olie-exporterende ontwikkelingslanden en IMF-beleid, 1973-1983	febr.
02. P. Kort	Aanpassingskosten in een dynamisch model van de onderneming	maart
03. G.J.C.Th. van Schijndel	Optimale besturing en dynamisch ondernemingsgedrag	maart
04. J. Kriens J.J.M. Peterse	Toepassing van de regressie-schatter in de accountantscontrole	mei
05. J. Kriens R.H. Veenstra	Statistical Sampling in Internal Control by Using the A.O.Q.L.-system (revised version of Ter Discussie no. 83.02)	juni
06. A. van den Elzen D. Talman	A new strategy-adjustment process for computing a Nash equilibrium in a noncooperative more-person game	juli
07. W. van Eijs W. de Freytas T. Mekel	Automatisering, Arbeidstijd en Werkgelegenheid	juli
08. A. van Soest P. Kooreman	Nederlanders op vakantie Een micro-economische analyse	sept.
09. H. Gremmen	Macro-economisch computerspel Beschrijving van een model	okt.
10. F. van der Ploeg	Inefficiency of credible strategies in oligopolistic resource markets with uncertainty	okt.
11. J. Moors	Some tossing experiments with biased coins.	dec.
12. F. van der Ploeg	The effects of a tax and income policy on government finance, employment and capital formation	dec.
13. C.P. van Binnendijk P.A.M. Versteijne	Stadsvernieuwing: vernieuwing van het stadhuis?	dec.
14. R.J. Casimir	Infolab Een laboratorium voor informatie-systemen	dec.

IN 1986 REEDS VERSCHENEN

01. F. van der Ploeg	Monopoly Unions, Investment and Employment: Benefits of Contingent Wage Contracts	jan.
02. J. van Mier	Gewone differentievergelijkingen met niet-constante coëfficiënten en partiële differentievergelijkingen (vervolg R.T.D. no. 84.32)	febr.
03. J.J.A. Moors	Het Bayesiaanse Cox-Snell-model by accountantscontroles.	maart
04. G.J. van den Berg	Nonstationarity in job search theory	april
05. G.J. van den Berg	Small-sample properties of estimators of the autocorrelation coefficient	april
06. P. Kooreman	Huishoudproduktie en de analyse van tijdsbesteding	april

Bibliotheek K. U. Brabant

17 000 01059165 0