Agricultural Economics, 5 (1991) 287–310
Elsevier Science Publishers B.V., Amsterdam

Concept and implementation of an integrated decision support system (IDSS) for capital-intensive farming

P. Wagner and F. Kuhlmann

Institute of Farm Management, Justus-Liebig-University, Senckenbergstraße 3, W-6300 Gießen, Germany

ABSTRACT

Wagner, P. and Kuhlmann, F., 1991. Concept and implementation of an integrated decision support system (IDSS) for capital-intensive farming. *Agric. Econ.*, 5: 287–310.

During the evolutionary process of developing software for management tasks, the need for integration became more and more obvious. This paper discusses how integrated information processing can be accomplished to support the managerial functions. Based on the concepts of control theory principal schemes of comparison possibilities and deviation analysis are shown. The philosophy behind the design of an integrated decision support system (IDSS), the on-farm implementation, and the integration problems of hardware and software are discussed. The applied IDSS consists of several planning and controlling models. These models and the linkages between them are described in detail.

1. INTRODUCTION

The need for more and better information on which to base decisions is not a new problem. However, in recent years, this problem has become even more important, particularly for capital-intensive farming in industrialized countries.

Information is required for different levels of farm management, ranging from very short-term decisions such as applying an insecticide, to very long-term decisions, such as building a hog-barn. In addition, the information needs for capital-intensive farming deviate from those of extensive farming. Capital-intensive farming is characterized by high sales volumes in comparison to the generated net value added, e.g. layer hens or feeder pigs where the monetary input is high in relation to sales volume, in contrast to range cattle where the monetary input is just a small part of the sales volume. Thus, in capital-intensive farming small changes in input–output

0169-5150/91/$03.50

coefficients and/or prices can cause net income to switch from positive to negative. Due to these facts, the inputs and outputs need to be monitored and controlled much more closely than in extensive farming. Therefore the information required for capital-intensive farming needs to be on higher levels with respect to quality as well as to quantity.

Providing farm managers with better information has been an evolutionary process. These efforts include developing electronic data processing systems, such as linear programming, management information systems (MIS) and currently, decision support systems (DSS). During this development process, the need for integration became more and more obvious:

- there are many occasions where information is needed to support a decision;
- there are many programs requiring more or less the same input data;
- the output of one program may be the input of others.

Because DSS's should have more emphasis on *human* effectiveness than on machine (computer) efficiency, ideally the data should be placed in one comprehensive data base which can be accessed by various models, which are placed in a model base. This conceptual design has been proposed by Sprague and Watson (1983, p. 22). As will be shown later, there are different ways from the idealized integrated DSS (IDSS) on the one side and stand-alone (independent) programs, on the other.

At this point some principal questions arise: If a user is able to use an IDSS and the relevant data are available for processing, he will get the information he is looking for. But how to continue when the information indicates that something is going wrong? What should be done next? How to take control? How to regain control?

This paper tries to answer those questions, therefore it first deals with the value of information and ways of processing and using it. Afterwards the role of integration will be discussed and finally the implementation of an existing IDSS on microcomputers on an experimental farm will be presented. The emphasis in this paper is placed on short-term controlling. Long-term controlling applications will be mentioned, but not be discussed in any depth.

The group of real-world decision-makers to which the described IDSS is addressed has been mentioned already: these are farmers practicing capital-intensive farming. Due to the fact that the principal planning and controlling needs for capital-intensive farming are almost always similar, the type of farm is not really important. That means that the IDSS can be used by a family-owned dairy farm, for example, as well as by more complex agricultural firms with employees and/or part-time workers. This is due to the nature of an IDSS: all of the models or a subset of them can be used independently. However, as a general rule it can be stated that the more intensive and diversified a farm firm, and the more profit centers it has, the more beneficial the use of an IDSS will be.

2. ROLE OF INFORMATION

2.1. Information

The more one knows about alternatives of actions, and their likely consequences and restrictions, the more successful one will be in general. In other words, the right information at the right time is the key to success. The basic problem, therefore, is scarcity of information. Information does not exist per se, information has to be produced. Information is obtained from data. The manager typically has a large amount of data, but a limited supply of information. As shown by Connor and Vincent [cited in Harsh, Connor Schwab (1981, p. 15)] information itself can be descriptive, diagnostic, predictive, and prescriptive.

2.2. Production and distribution of information in a firm

In systems theory, firms can be defined as open dynamic systems (Baetge, 1974, p. 11). For the purpose of this paper they can be characterized further by splitting them into a basic subsystem and an information subsystem, as shown in Fig. 1 (Kuhlmann and Wagner, 1986, p. 410). The information subsystem refers to the basic subsystem. Figure 1 shows that the basic and

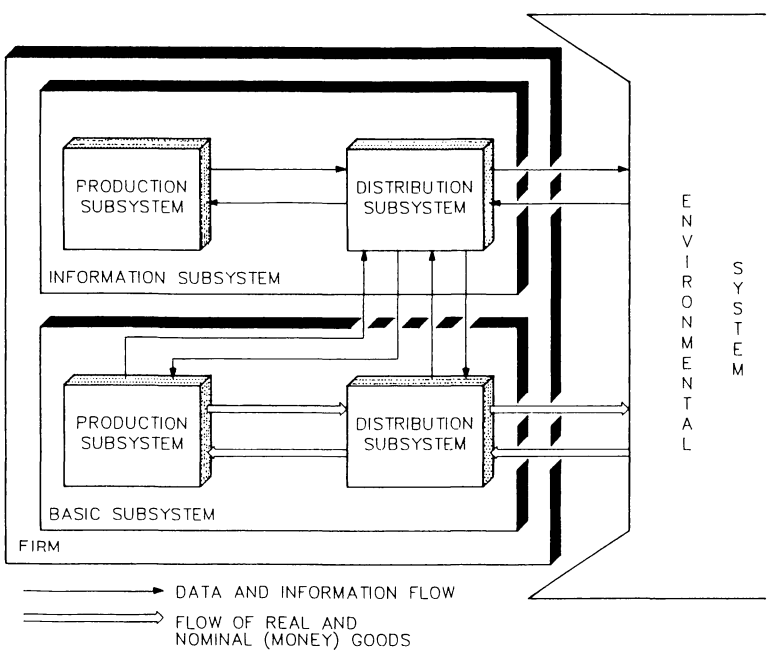

Fig. 1. Information subsystem and basic subsystem of a firm.

the information subsystem can both be subdivided into a production and a distribution subsystem. The two distribution subsystems connect the production subsystems with the environmental system. The basic subsystem represents those parts of a firm where real and nominal goods are transformed. Here, the production inputs are purchased, shared and transformed into products. The products are sold. The flow of nominal goods (money) is induced by these processes. In the basic production subsystem the production of real goods takes place. The basic distribution subsystem keeps contact with the environmental system: goods are transformed over distance and time.

The information subsystem behaves in analogy to the basic subsystem. The information distribution subsystem first takes or receives data as informational production factors from the environmental system, the basic production subsystem and/or the basic distribution subsystem. The data then are stored or transmitted to the information production subsystem, where they are processed into information. The information gained is transmitted to the environmental system, e.g. in form of orders, advertisements, etc., or to the basic subsystem, e.g. in form of instructions and results.

The information subsystem of a firm is the place where the DSS is located. The information distribution subsystem holds the database and the information production subsystem holds the model base. Therefore the information subsystem will be the further object of consideration.

3. SUPPORTING MANAGERIAL FUNCTIONS BY INFORMATION PROCESSING

3.1. Management functions in a firm

Goal-oriented management needs to use the cybernetic concepts of open- and closed-loop control. Prior condition for the use of those concepts is the installation and application of decision-support systems. The DSS contains models of the system or models of parts of the system, where the system is the firm to be managed. However, the concepts of open- and closed-loop control shall not be discussed here, because this material can be found in other publications, [e.g. Kuhlmann and Wagner (1986, p. 413) or Kuhlmann, Berg and Harsh, (1984, p. 21)].

3.2. Managerial process in a firm

The managerial process may be subdivided into six subprocesses:

(1) definition of goals
(2) planning (observation and analysis)

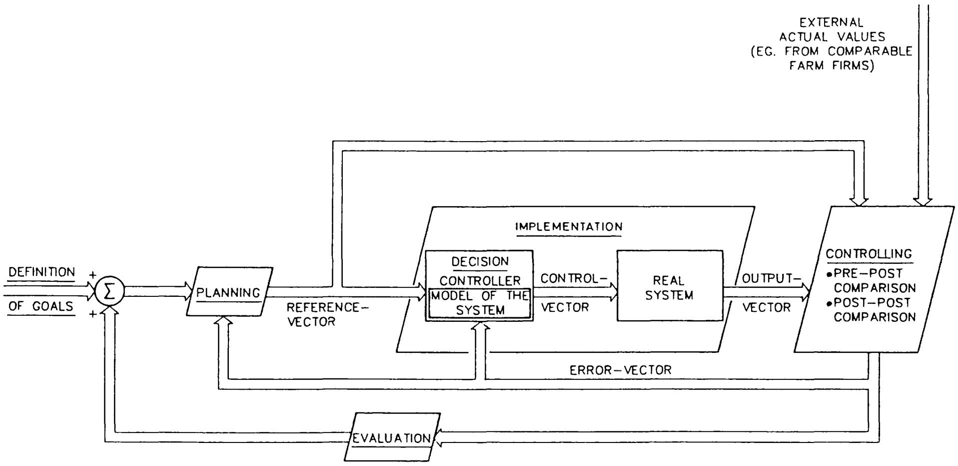

Fig. 2. Controlling as function of operating a firm.

(3) decision
(4) implementation
(5) controlling
(6) evaluation.

Other studies may give other definitions of the managerial process, depending on the theoretical point of view. In order to define the managerial process according to an IDSS, the above definition is sufficient.

The six steps of the managerial process are to be understood as an iterative process rather than a one-way sequence, as shown in Fig. 2.

The process starts with the definition of goals. The definition of goals has a major influence on the outputs of the planning process. The plan defines the objectives to be realized by the system, for which the reference vector must be transferred into the control vector. The real system, now acting with the control vector, delivers actual values (output vector) after a certain time lag. Those values then need to be compared under different aspects, such as preset objectives by the means of a pre-post comparison or any other, e.g. external values by a post-post comparison. Probably there will be some deviation or error. The error vector itself provides three different kinds of information:

(1) It will tell the controller (the model of the system) that something went wrong. The controller has to decide what to do to bring the real system back on course: the controller must define a new control vector.

(2) In the case of fatal errors or long-lasting errors of the same direction, the plan should be considered for correction, because defined objectives could have been wrong.

TABLE 1

Actors in the phases of management depending on structural optimization or process optimization

Phases of management	Action taken by	
	Structural optimization	Process optimization
Definition of goals	manager	manager
Planning	manager/model	manager/model
Decision	manager/model	manager/controller
Implementation	manager	manager/actor
Controlling	manager/controller	manager/controller
Evaluation	manager/controller	manager/controller

(3) The errors must be evaluated to learn what the reasons for the errors were. This may happen periodically and is the case particularly if actions 1 and 2 have not led to the desired results. The evaluation may show that the goals have not been well defined. Either the reference vector turned out to be miscalculated or the goals are simply not realizable with the given assets or production capacity.

It has to be said at this point that the values of the reference vector are future values. They can only be estimated and not predicted with complete certainty. However, there are ways to compute those values with a higher certainty than it has been done in most cases in the past, as will be shown in Section 5.

In the decision and the implementation steps there are some differences concerning the actor. This depends on whether to optimize the *structure* or a *process* of a firm. Table 1 shows this setting.

3.2.1. Structural optimization versus process optimization

The optimization of structures in general is a design problem and has two dimensions. The first one cannot be directly influenced by the manager, for example new varieties of cash crops, which require fewer inputs or have higher yields, and other factors which are influenced by technical or biological progress. The second dimension, which Table 1 refers to, can easily be influenced by the manager, e.g. the structure of a firm, crop rotation, the variety of dairy cows. These are factors which 'have to be lived with', at least for a certain period of time. This does not mean that there is no room for corrections or changes, but the decision on a certain crop rotation can only be changed when the period of vegetation is over, for example.

The nature of process optimization is a different one. The process is predefined by the structure, but the actions to be taken in the process may vary and can be adapted if necessary, often very quickly in very short periods of time. Feeding programs for pig fattening may serve as an example.

In a complete IDSS, models must be implemented to manage both problems. Referring to Table 1, the two examples above shall be taken to illustrate by whom actions may be taken in order to optimize structures or processes.

The optimal configuration of a firm for example can be found by a simplex algorithm or just by trial and error (simulation). In any case the manager has to define a goal, e.g. maximize the overall gross margin. The planning phase can be accomplished by the manager himself or by any LP program. In the latter case the result is normative, there is no decision because the information is prescriptive. In the simulation case the manager has to decide what to do. The implementation of a chosen production program is left to the manager. He, or a controller (which would be a computer program), may then control the success of the implementation of the plan by comparing the results with the predefined objectives. Finally, the obtained results have to be evaluated, either by the manager himself or by a program designed for this purpose.

In the second example (pig fattening) the goal again must be set by the manager. A possible goal could be increasing each pig's weight by about 700 g per day. How to reach that? Again, this can be planned by a computer program or by the manager. In any case decisions have to be made about the feed ratio (in the first step), the combination and the amount of ingredients, or changing of a ratio (in the second or any further loop in case of error). The adjustment of ratios, i.e. computing new ratios, can be performed by a controller (model of the system). The implementation can be accomplished either by a feeding computer (actor) or by hand (manager). The controlling as well as the evaluation phase can be depicted as in the first example: The manager or the controller may control the success of the implementation of the plan by comparing the results with the predefined target values of his own farm and/or actual values from successful comparable pig-fattening farms. The error values which are needed to adjust the ratios are then generated. The two facts, that process optimization and adjustment can happen very quickly, and that decision *and* action can be done by machines, show that process control and optimization can be accomplished much easier automatically than structural optimization.

3.2.2. Pre-post comparison versus post-post comparison

In the following the concepts of pre-post and post-post comparison as the two major controlling tools of Fig. 2 will be explained in detail. The principal scheme of both is shown in Tables 2 and 3.

TABLE 2

Principal schemes of comparison possibilities: pre-post comparison

	Sum	January	February	March	April	May	...	December
efficiency ratio 1								
target value (C)	2400	200	400	600	800	1000	...	2400
actual value (C)	780	180	340	540	760	x	...	x
deviation (A)	1620	20	60	60	40	x	...	x x
deviation (R)	−67.5	−10.0	−15.0	−10.0	−5.0	x	...	x
efficiency ratio 2								
⋮								

(A), absolute; (C), cumulative; (R), relative.

The pre-post comparison compares the development of several defined efficiency ratios over time. The objectives (target values) originate in the planning process. In the example they are generated in monthly steps at the beginning of the year. The actual values have to be put in whenever they occur. Hence, the comparison shows the absolute and relative errors per month and to what degree the plan is fulfilled. Pre-post comparisons indicate deviations at an early stage, and thus permit early corrective actions. The values may be accumulated over time to get smoother time series.

The post-post comparison is used to provide information about the result of managing a firm in relation to *comparable* firms, such as farms of the same region and the same kind of production system (horizontal comparison). This type of comparison should explain the reasons for deviations. The corresponding data of other comparable farms (reference values) may be

TABLE 3

Principal schemes of comparison possibilities: post-post comparison

	Reference values						Actual values	
	unsuccessful farms		average farms		successful farms		farm to compare	
	(A)	(R)	(A)	(100%)	(A)	(R)	(A)	(R)
efficiency ratio								
1	400	80.0	500	100.0	550	110.0	530	106.0
2	1900	95.0	2000	100.0	2200	110.0	2008	100.4
3	360	102.9	350	100.0	325	92.9	350	100.0
⋮								

(A), absolute; (R), relative.

collected by the extension service or by accounting bureaus. The time horizon of such a comparison may be a year.

These two ways of comparing data are only examples, though there are many other ways of doing this job, e.g. time-series analysis of several efficiency ratios of one firm (vertical comparison). This may show the firm's development path.

The basic idea of such comparisons is to learn. The learning process starts with recognizing a deviation. It continues by analyzing the causes of the deviations (the evaluation step) and by learning how to do it better.

3.2.3. Deviation analysis by means of controlling

Basically, deviations between the desired and the actual state arise from by three causes; these are deviations in *quantity, price* and *structure*. In addition, each of these three different sources of deviation may be caused by internal or external factors, i.e. controllable or noncontrollable variables. To bring the system back on course, the reasons for the deviation must be analyzed first. In gneral, short-term adaptation can be accomplished by trying to influence quantities and/or prices, whereas in almost all cases long-term adaptation stands for manipulating the structure of a firm.

However, when emphasis is placed on short-term process control, price and quantity effects must be considered first. Analyzing gross margins may serve as an example. A procedure of analysis is shown as a flow chart in Fig. 3. If, as shown in the example of Fig. 3, deviations between actual gross margins and target gross margins occur, one has to find out first whether they are caused by prices (or quantities, the order is not important at this point). In order to deterine the reasons for price deviations, the actual prices of the considered farm have to be compared with the target prices as well as with the reference prices, which may be obtained from comparable farms. A similar procedure may be necessary to repeat for the quantities.

3.2.4. Ways to help entrepreneurs in decision making

If the farmer is faced with the problem of taking actions, he needs to decide what to do and what not to do. Therefore he needs the ability to perform his managerial tasks efficiently. There are commonly four methods of improving that ability [Öhlmér and Nott, cited in Polyakov, Kuhlmann and Öhlmér (1981, p. 103)]:

(1) Providing the farmer with information about relevant data (e.g. available facilities and services), about problematic situations and about analysis and planning methods. This kind of help utilizes written material and broadcasting which is directed to many farmers.

(2) Increasing the farmer's knowledge and managerial skills, so that he will be able to perform the management task on his own. This means

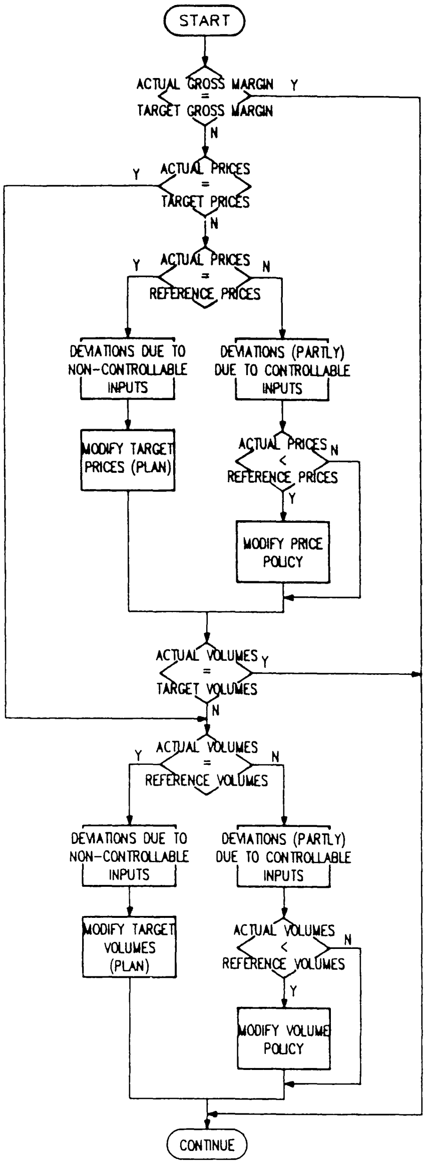

Fig. 3. Analyzing volumes and prices.

education about the situation (problem), relevant information, analysis and planning methods, available facilities and available services. Each activity within this kind of help is directed to a group of farmers.

(3) Face-to-face service, i.e. an extension or commercial agent helps the farmer in doing a part or all of the managerial tasks.

(4) Providing facilities which the farmer can use by himself and then be able to perform the management task. Each of these facilities is used by a single farmer, although the facilities can be mass-produced.

The implementation and usage of an IDSS is grouped under point (4). This does not mean that all problems are solved only by implementation of a system. This can be only one step in the process of improving the farmer's ability to perform his managerial task. In order to get some benefits out of an IDSS the farmer needs to be educated and informed. Using an IDSS requires knowledge about the implemented methods and tools, and about the quality of data available and the capability to evaluate the results. On the other hand, it may be easier to perform the managerial task with the support of an IDSS. A modern IDSS can help by selecting and using the appropriate tools and methods for a given problem as well as by interpreting the results and recommending the actions to be taken. This will be discussed in Section 5, where the general structure of an IDSS will be presented.

4. ROLE OF INTEGRATION

Before an implemented IDSS will be described in Section 5, it shall be discussed why a DSS needs to be a somehow integrated system and what integration means in the context of a DSS. Integration of information processing can be seen at least at two basic levels: hardware integration and software integration.

The integration of hardware deals with compatibility and is mandatory for integrating software. In regard to hardware integration, most problems are of technical nature, such as the design of interfaces, the kind of handshake or just the compatibility of magnetic tapes or, more important nowadays, disk formats and sizes. Therefore, hardware integration problems arise if computers need to be connected. There is no need to worry about integrating hardware if all the software used is to run on the same machine and the data are entered into the computer via keyboard. However, this is impracticable, as the later example will show, because a substantial portion of the needed data are registered via sensors and stored in devices, which may be called 'process control computers'. This will become even more important in future capital-intensive farming.

The second level of integrating mentioned above is software integration. At least three different sublevels are to be considered here; the enumeration follows in ascending order of practicability and ascending level of possible complications:

(1) No direct linkage between programs. The output of one program must be reentered in another program via keyboard. Integration here means

matching units, e.g. output in hectares of one program cannot be used as input in acres in the other program. Aside from this there will be no particular problems.

(2) Indirect linkage between programs. Data between programs can be transferred via specific connection modules (these are programs, too), but at this level there is no common data which would be accessible to the communicating program. The communication needs to be performed by file operations. Problems arise if programs or data structures need to be modified, if, for example, the units do not match as described above, and last but not least if the process of transferring data from one program to another is not automated, the user may simply forget to start the transferring program before doing some analysis with the program which needs to receive the data.

(3) Automatic linkage of all programs. That means, that any input is to be entered exactly on time. Every other program requiring the same input will check in a common database if the data are already entered. Output is written into the database as well, so that other programs can refer to the results of preceding ones. The problem at this level of software integration is to be seen in defining the data structures and managing the database. Even more complicated is how to 'tell' each individual program whether to ask the user for data or to look in the database first, and, if the data were found in the database, to decide whether they meet the needs of the special application the user wants to run. Due to the fact that software is not 'static' (this holds true for IDSS, too), new programs may be added, old ones may be dropped or existing ones changed. Some cases may require independent usage of a program. All these things have to be managed by the data base and the model base manager (see Section 5 for explanation) automatically. It cannot be managed by the IDSS user.

This last and most sophisticated level of integration is the goal to be achieved if an IDSS is to become user-friendly and widely accepted. IDSS is not at this level of sophistication of IDSS at present.

5. CONCEPT OF AN IDSS IN AGRICULTURE

5.1. General structure of an IDSS

The general structure and the components of an IDSS are described in Fig. 4. This theoretical approach is presented by Sprague and Watson (1983, p. 22) and modified here. The major components are the data base, the model base and the user support base. In addition, but not less important, there are management systems for all three bases, a user interface and the decision-maker himself. Each of the components will be examined independently.

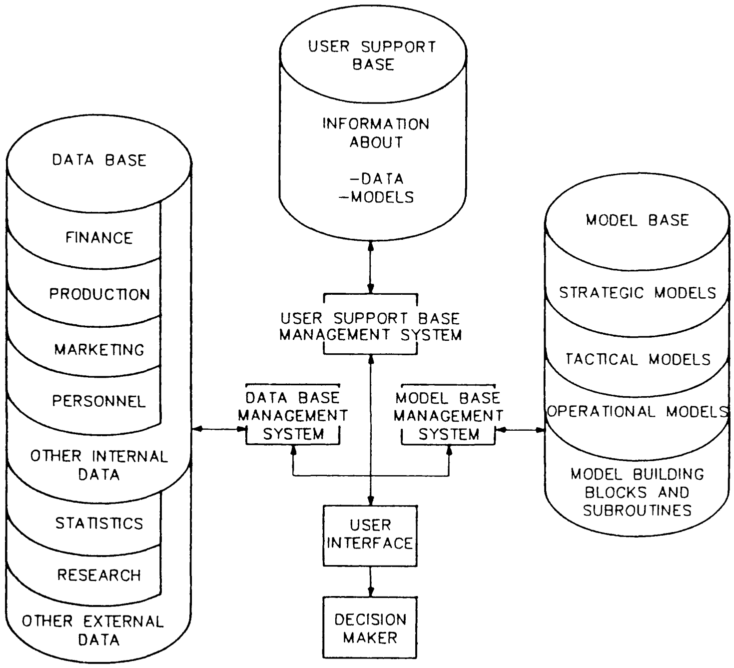

Fig. 4. General concept of an IDSS. Source: Sprague and Watson, 1983, p. 22 (modified).

Database and database management system. A database system is used to store classes of data which have been collected for various purposes such as financial data, production data, statistical data, and so forth. The data can be generated from the firm itself or from external sources. Because the sketched IDSS meets the conditions described as the third level of integration, the various databases need to be consistent within the overall structure and need to be shared across functional needs. This means, for example, that the accounting data is not stored using a different system than the production or statistical data.

Model base and model base management system. The model base is closely connected to the database. The model base contains several kinds of models, some of which are used for strategic planning (structural optimization), others for supporting tactical and operational decisions (process optimization). The model base management system performs the same basic tasks as the database management system. It is charged with retrieving the appropriate model needed for a specific purpose and then requesting the necessary data for the model from the database management system and/or the user interface.

User support base and user support base management system. The user needs to get information about the data/database and/or models contained in the model base. That information is kept in the user support base, which could be called 'information base' as well, but the word 'Information' is already used in another sense (see Section 2.1). The user-support base management system, for example, could be a powerful hypertext system. Besides the model base, this is the most important part of an IDSS when it comes to user acceptance. Not every user knows about the possible applications of every model contained in the model base. Of course, there are other ways to help the decision-maker here, as depicted in Section 3.2.4, but in the short run the user support base *must* contain all necessary information about the usage of models and data of the IDSS. The relevant models for short-term controlling are contained in the operational part of the model base. Therefore, the emphasis on the following description of an implemented IDSS will be concentrated on this part.

5.2. Concept of the applied IDSS

The design of the applied IDSS is illustrated in Fig. 5. The IDSS has been developed at the Institute of Farm Management at the University of

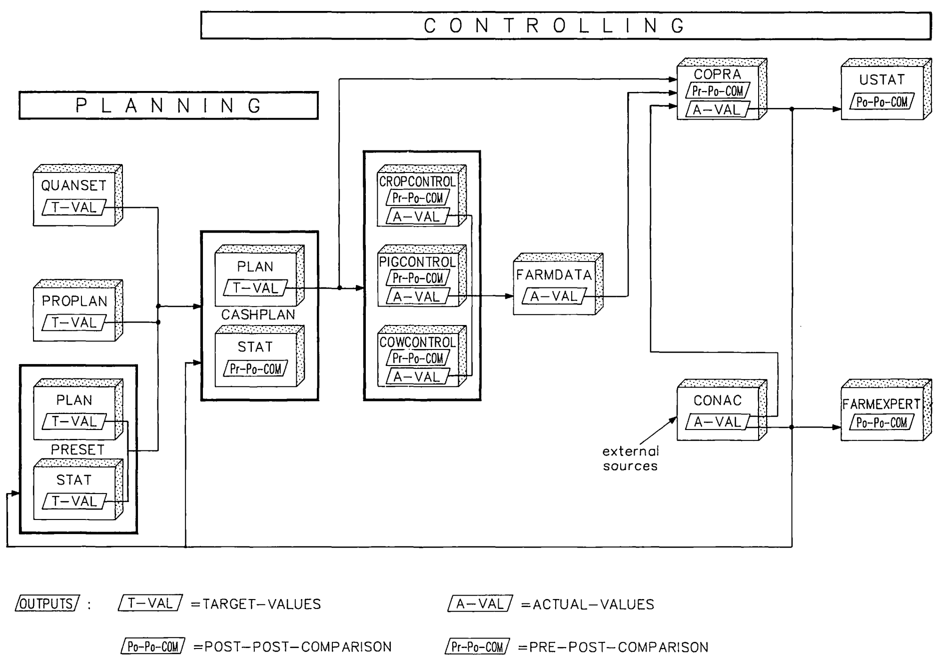

Fig. 5. Concept of the applied IDSS.

Giessen. It presently is implemented and tested on the experimental farm of the Institute, Marienborn.

The models can be subdivided into two major groups: planning models and controlling models. The planning models are generating the reference (target) values, the controlling models are processing the actual values and the pre-post and post-post comparisons, respectively.

In the following the linkage of the models will be discussed first, then the models themselves.

Planning models. QUANSET generates the reference values for the quantities, PRESET those for the prices, and PROPLAN those for the structure of the firm. All three deliver their generated data to CASHPLAN; CASHPLAN stands for cash flow planning and control.

Controlling models. CASHPLAN is both a planning and a controlling model. The defined efficiency ratios can be compared by means of a pre-post comparison. It provides the reference values for CROPCONTROL, PIGCONTROL and COWCONTROL. The three programs control the crop production, pig fattening and milk production. They give their actual values to FARMDATA, a program which provides an overview over everything that happens on the farm, except financial transactions. Those transactions are handled by CONAC, a book-keeping program. CONAC is connected to PRESET and CASHPLAN. PRESET needs past time series of prices for the prognosis of future prices (reference prices), CASHPLAN gets the actual values for the pre-post comparison from CONAC. COPRA, a cost accounting program, gets the reference value from CASHPLAN and the actual values from FARMDATA (volumes) and CONAC (prices). It provides a pre-post comparison for every single production process. USTAT and FARMEXPERT compute post-post comparisons. They compare special efficiency ratios of the farm with those of comparable farms.

5.3. Physical linkage of the models

The above outlined linkages are of logical nature, the physical linkages describe how the data actually are transferred from one model to another. The transfer between the planning programs is via keyboard. That means the outputs of QUANSET, PRESET and CASHPLAN are printed on paper. The figures as input for the connected programs have then to be re-entered via keyboard by the user. This seems reasonable, because the target values have to be generated just once a year. The linkage between FARMDATA, COPRA and CONAC as well as USTAT and FARMEXPERT also has to be performed manually. There is no other way, so far. It would be worth thinking about a

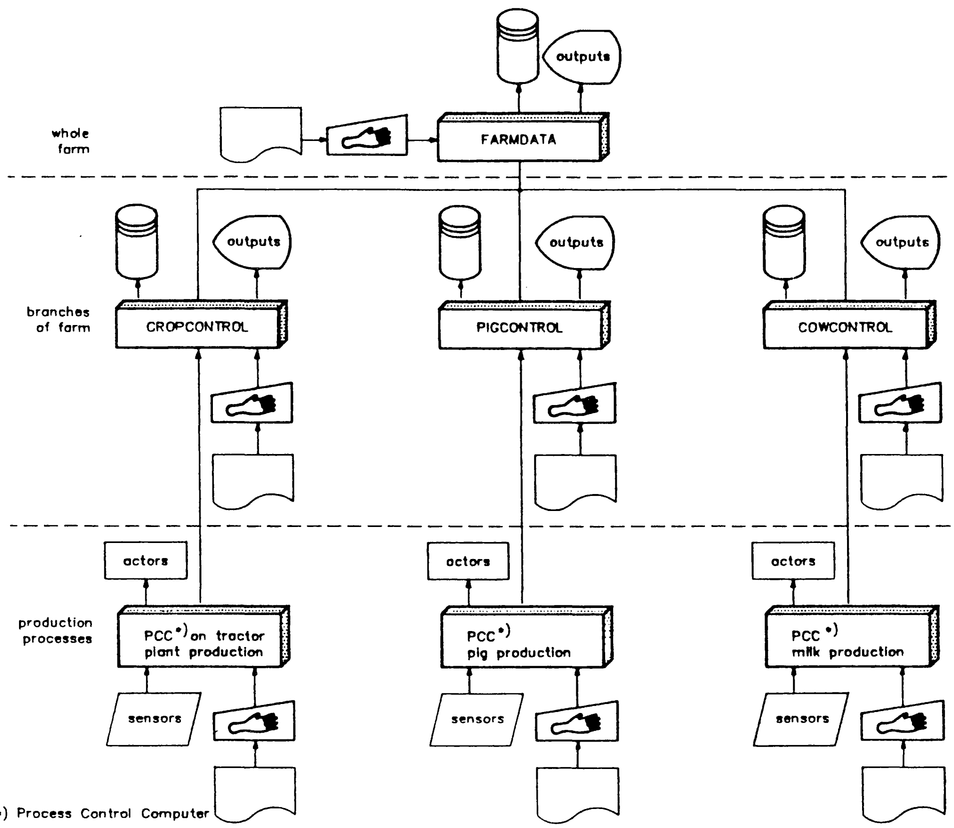

Fig. 6. Information linkages between the farm levels.

transfer program, because the amount of data to be plugged in is fairly large. The most challenging task, however, is the linkage between the programs which require actual data as input, that is to say the programs which stand in close connection to the real world (the production processes). Gathering and entering all of the data from the production processes by hand would make the programs unacceptable and would be too labor-intensive. So, this process has to happen automatically. How the linkage between the process computers, and the programs CROPCONTROL, PIGCONTROL, COWCONTROL and FARMDATA has been realized, is shown in Fig. 6.

At the level of the three production processes, control is left to the process-control computers. The computers are responsible for feeding the pigs and dairy cows. Furthermore, the process-control computers gather the data via sensors, such as the daily milk production or how much of fertilizer has been used on the fields that day. Some of the data, the names of the fields, for example, have to be entered via keyboard.

The process-control computers are connected via cable to the process-management computer. At this level a lot of software and especially hardware integration problems arise during installation.

The bottom-up link between the process management computers on which the three controlling programs are installed and the farm-management computer with FARMDATA is via diskettes. This could be replaced by a network as well. The top-down link, that means the connection of CASHPLAN and the controlling programs, is via keyboard, because, as already stated, the reference values for the controlling programs only have to be entered once a year.

Finally, on the farm-management computer the files of the control programs are joined by a transfer program which makes the data compatible to FARMADATA.

5.4. Description of the models contained in the model base

In the following paragraph each of the models contained in the model base of the IDSS will be presented. The widely used programs such as CONAC, a double-entry bookkeeping program, are just mentioned in order to sketch the whole system, but are not described in any detail.

5.4.1. Models for planning purposes

(1) QUANSET

QUANSET is a program to generate quantities. The basic idea is, that for planning purposes one needs detailed information about the amount of inputs for the production processes to be planned. The conventional way is to get those values from tables or to simply estimate, how much of fertilizer, seeds or working hours, for example, are needed to produce wheat, barley, sugar beets on 1 ha, or what kinds and amounts of inputs are necessary to produce 400 000 liters of milk or 2000 fattening pigs per year.

QUANSET creates such figures for the user. The user has to define a desired production level for the particular process to be planned, e.g. winter wheat. That could be about 5, 7 or 9 t/ha. The program then will come up with a proposal of required inputs. The user now can modify the suggestions according to specific circumstances of his farm or just accept the given values. It is importanat to note that the proposal includes the points of time when the inputs have to be applied.

---

t, metric tonne = 1000 kg.

(2) PRESET

The counterpart of QUANSET is PRESET. PRESET forecasts prices for inputs and products, based on past time-series by means of a method called 'Adaptive Filter' (Röhrig, 1989, p. 69). Thus, PRESET comes up with a prognosis for the price per unit for the desired product or production factor in monthly steps for about one year in advance in the same fashion as QUANSET. The user may accept the production or correct the prices, if he feels that something could happen that would influence the prices in a different direction. Either way, the information is of a predictive nature (see Section 2.1). The program also compares farm prices with market prices in its statistical part to show deviations; in other words it analyzes price effects in detail (see Section 3.2.3).

(3) PROPLAN

PROPLAN basically is a linear programming model based on gross margins. It generates the reference values for the structure of the farm. That means how many hectares of wheat, barley or corn should be in the crop rotation, how many dairy cows are most efficient, etc. Joining the quantities of QUANSET, the prices of PRESET and the optimal farm structure obtained from PROPLAN gives a complete preliminary budget for the farm; in other words, the plan for the upcoming production period.

(4) CASHPLAN

The necessary combination of structure, quantities and prices is accomplished by the planning part of CASHPLAN. Here, the projected prices per unit and the quantities per unit are matched and calculated for a complete production budget by multiplying them with the farm's resources. The budgets can be defined for all production processes of the farm. The production budget is projected for about one year in advance in monthly steps. Multiplying prices, quantities and the resources (described by the structure of the farm) on a monthly base yields monetary inflows for sold products and monetary outflows for purchased production factors over time. Thus, control over liquidity is maintained. At the last step, an expected balance sheet and profit-loss account is generated. The results of CASHPLAN show what, when, where, how and by whom to do. The reference vectors are set by this. Thus, CASHPLAN generates prescriptive information (see Section 2.1).

5.4.2. Models for cost-performance control and pre-post control

(1) CASHPLAN

CASHPLAN delivers a pre-post comparison of free definable efficiency ratios on farm level in monthly steps. The target values are generated in the

TABLE 4 Selected figures of a pre-post comparison by CASHPLAN, Marienborn, Planning period: 01.01.89–31.12.89

Formula 3 label: *Gross income of farm*												
target value (C)	112460	191660	300100	480520	602880	688080	782620	916180	1027015	1118190	1222985	1378210
actual value (C)	93092	192055	265050	409899	516735	649940	757926	887563	968318	1152901	1152901	1152901
deviation (A)	−19368	395	−35050	−70621	−86145	−38140	−24694	−28617	−58697	34711	−70084	−225309
deviation (A) (relative)	82.78	100.21	88.32	85.30	85.71	94.46	96.84	96.88	94.28	103.10	94.27	83.65
Formula 5 label: *direct expenditures*												
target value (C)	54096	130359	197789	271978	337552	399938	456725	503661	553237	616905	675775	721640
actual value (A)	20427	98559	154141	244271	312690	354105	417039	478046	548320	627610	627610	627610
deviation (A)	−33669	−31800	−43648	−27707	−24862	−45833	−39686	−25616	−4917	10705	−48165	−94030
deviation (R)	37.76	75.61	77.93	89.81	92.63	88.54	91.31	94.91	99.11	101.74	92.87	86.97
Formula 6 label: *Overhead expenditures*												
target value (C)	17350	30700	42050	62900	74650	108000	121350	146700	164050	175800	191050	234800
actual value (C)	10279	22688	38892	67473	85976	112564	148201	167462	188642	219153	219153	219153
deviation (A)	−7071	−8012	−3158	4573	11326	4564	26851	20762	24592	43353	28103	−15647
deviation (R)	59.24	73.90	92.49	107.27	115.17	104.23	122.13	114.15	114.99	124.66	114.71	93.34
Formula 8 label: *Wages*												
target value (C)	21760	42533	66131	85632	107948	127764	146343	164263	190438	213505	244985	265505
actual value (C)	24385	44166	61593	80053	100663	121821	149330	165018	184965	216177	216177	216177
deviation (A)	2625	1634	−4537	−5579	−7285	−5944	2987	756	−5472	2672	−28808	−49328
deviation (R)	112.06	103.84	93.14	93.49	93.25	95.35	102.04	100.46	97.13	101.25	88.24	81.42

(A) absolute; (C) cumulative; (B) relative planning module of CASHPLAN as described, the pre-post comparison is computed in the statistical part of it. The actual values for that comparison are the results of FARMDATA, COPRA and CONAC. Table 4 shows, as an example, some selected efficiency ratios of a pre-post comparision by CASHPLAN. The figures represent the actual state of the planning period until October. Because the figures are cumulative (see Section 3.2.2) it can be seen at a glance how much is left to reach the final target at the end of the planning period. In order to analyze the reasons for registered deviations, many other efficiency ratios have to be considered, which are provided by CASHPLAN as well. Thus, as already mentioned in Section 3.2.3, each deviation of quantities and prices becomes obvious, and the reasons for deviations in more aggregated efficiency ratios and can be retraced from here.

(2) *Process control programs*

All three of the following control programs provide detailed pre-post comparisons. Each of those comparisons consists of comparison of quantities and costs separately. This enables the user to filter effects caused by quantities and/or prices in the case of deviations. The three controlling programs produce descriptive information (see Section 2.1).

(a) CROPCONTROL

CROPCONTROL is a program for managing crop production during the vegetation period; it is an enhanced computerized field record system. CROPCONTROL contains:
- fertilizer balancing
- fertilizer planning
- comparing of crop rotations
- comparing of crop types
- comparing of "on-farm" experiments
- comparing of seed varieties
- cost and result accounting
- pre-post comparison (Seck, 1988, pp. 94 ff.).

The actual data as input for CROPCONTROL have their origin directly from the field; the target values are provided by CASHPLAN and may be specified for the controlling purposes of the program.

(b) PIGCONTROL

The management of feeding pigs is the objective of PIGCONTROL. With PIGCONTROL the farmer is able to keep track of every group of pigs from the time they get into the barn until they leave. It includes:
- feed ratio optimization

- cost and result accounting
- pre-post comparison (Lang, 1989, pp. 35 ff.).

The actual data for each group are stored in the feeding computer and transferred daily to the process management computer. The target values, again, are provided by CASHPLAN and may be more clearly specified for the controlling purposes of the program.

(c) COWCONTROL

This program helps manage milk production. For this purpose each dairy cow is individually registered in the program and the process-control computer, which automatically transfers the actual data to COWCONTROL. Each cow wears an identification tag, so that the feeding ratio can be determined with respect to the individual performance of the cow. In the same manner, each cow's daily milk-production figures are stored by the computer. This makes it possible to obtain:

- veterinary analysis for each cow
- cost-type accounting
- cost-center accounting
- cost-unit accounting (for each cow)
- ex-post cost and result accounting for all cows
- pre-post comparison (Müller and Kübler, 1989, pp. 105 ff.).

(3) FARMDATA

FARMDATA is the aggregation module of the actual values delivered by CROPCONTROL, PIGCONTROL and COWCONTROL. The values of the single production processes are brought together on the farm level. FARMDATA itself provides the aggregated data for COPRA. The most important outputs of FARMDATA are:

- inventory bookkeeping
- labor diary and -accounting (Wagner and Langenbruch, 1987).

(4) COPRA

COPRA as general cost and result accounting contains:

- cost-type accounting
- cost-center accounting
- cost-unit accounting
- cost-unit-period accounting based on direct costs
- cost-unit-period accounting based on full costs (Wagner, 1983, pp. 149–158).

(5) CONAC

CONAC is a double-entry bookkeeping program. Here the realized 'on-farm' prices are stored for later use by PRESET.

5.4.3. Post-post-comparison models

(1) USTAT

USTAT and FARMEXPERT are the programs for comparing a considered farm with other farms, but in a very different way. USTAT furthermore enables the user to compare his farm vertically. The program uses free definable efficiency ratios to compare the entire farm, production branches or whatever the user is interested in. Preconditions for the horizontal farm comparison are external data, at least at the level the user wants to compare his farm with. Thus, the major outputs of USTAT are *horizontal* and *vertical comparisons* for the farm. The external data have to be chosen and entereed by the user; the farm data are delivered by CONAC, COPRA and FARMDATA.

(2) FARMEXPERT

FARMEXPERT is an expert system. Its only output is a *horizontal farm comparison*. In contrast to USTAT, the outputs here are not just raw figures. Instead of simply producing statistics, FARMEXPERT analyzes deviations and explains the reasons for the deviations. The program analyzes retrospectively price, quantity and structural effects, as they are mentioned in Section 3.2.3. Furthermore, it contributes conclusions about the profitability of the farm.

The outputs generated by FARMEXPERT are an example for diagnostic information (see Section 2.1).

The data of the comparable farms are not to be entered by the user; they are provided by a database according to comparable regions and farm types, so that the farm under consideration can be compared with a group of similar farms.

The actual data of the farm under consideration are provided, as in the case of USTAT, by CONAC, COPRA and FARMDATA.

6. CONCLUSIONS

The presented IDSS is the result of rather comprehensive research activities, carried out within the last three years. This holds true especially with respect to planning models and physical integration. Due to that fact, the IDSS depicted is presently being used and tested on just one farm. In general, the appropriateness and efficiency of the applied IDSS shows encouraging results. On the other hand it had to be recognized soon that the amount of data to be entered via keyboard is unacceptable for practicing farmers. Nonetheless, some of the models such as PROPLAN or CASHPLAN (150 copies sold within 18 months) are well accepted by the farmers as stand-alone programs. Obviously, the reasons are:

(1) The two programs represent well-known approaches such as linear programming (PROPLAN) and budgeting (CASHPLAN). Therefore, it is possible for the user to reason about the expected benefits beforehand.

(2) PROPLAN and CASHPLAN are to be used for planning purposes. There is no need to run them every day; the user may decide when and how often to use them. This is different from the controlling programs, where data have to be entered almost every day. For those cases where not even a connection to a process control computer is available, the time for entering the data is seen as not acceptable by most farmers. This is even more of a problem, as the final reason of all the efforts is cost-accounting, an approach which is not widespread and accepted by farmers so far.

(3) The interdependences between so many programs as in the above described IDSS are of a very complex nature. Thus, the 'black box' simply becomes too big for the user to accept without any doubts. Therefore, many users prefer stand-alone programs.

To enhance the acceptability of the IDSS, as well as parts of it, at least three things should be done:

(1) Despite the level of integration already achieved, the need for better linkages between the single programs is obvious. We are working on it.

(2) Apart from the model compilations, the selection of appropriate efficiency ratios for pre-post comparison and post-post comparison has to be accomplished carefully to avoid information overkill as well as insufficient information.

(3) The farmers' knowledge and managerial skills have to be improved, as already mentioned in Section 3.2.4. This can be done by extension services or, even better, in earlier stages, e.g. vocational training institutions and universities. Therefore we are providing those institutions with our IDSS or parts of it in order to make it available for educational purposes. In this case we fill in the data base with default data, so that the students will enjoy using the programs and approaches behind them, without being trapped by frustration due to the boring job of entering data. In this way it is possible to increase the transparency of existing problems and to show how to solve them by using an IDSS.

Recent discussions with farmers showed that there is an increasing sensibility and awareness about the upcoming techniques of information processing by means of IDSS's.

REFERENCES

Baetge, J., 1974. Betriebswirtschaftliche Systemtheorie. Westdeutscher Verlag, Opladen, 277 pp.

Harsh, S.B., Connor, L.J. and Schwab, G.D., 1981. Managing the Farm Business. Prentice-Hall, Englewood Cliffs, NJ, 384 pp.

Kuhlmann, F. and Wagner, P., 1986. Zur Nutzung der Informationselektronik. Ber. Landwirtsch., 64: 408–440.

Kuhlmann, F., Berg, E. and Harsh, S.B., 1984. On decision support systems using adaptive control procedures. In: C.H. Hanf and E. Schiefer (Editors), Agricultural Markets and Prices. 4th European Congress of Agricultural Economists, 3–7 September 1984. Vauk, Kiel, Germany, pp. 13–34.

Lang, N., 1989. Pigcontrol. Gießener Schr. Agrar-Ernährungswirtsch. 18, 251 pp.

Müller, H. and Kübler, H., 1989. Computergestützte Steuerung und Kontrolle in der Milchviehhaltung. Ergeb. Landwirtsch. Forsch. Justus-Liebig-Univ. Gießen, 19: 105–110.

Polyakov, M., Kuhlmann, F. and Öhlmer, B., 1981. Computerization of farm management decision aids. In: G. Johnson and A. Maunder (Editors). Rural Change: The Challenge for Agricultural Economists. Proc. 17th Int. Conf. Agricultural Economists, 3–112 September 1979, Banff, Alta., pp. 102–112.

Röhrig, C., 1989. Preset. Ein computergestütztes Modell zur Dokumentation, Prognose und Kontrolle von Faktor- und Produktpreisen. Dissertation, Gießen, 216 pp.

Seck, M., 1988. Zur computergestützten Führung in der Pflanzenproduktion. Gießener Schr. Agrar- Ernährungswirtsch. 16, 183 pp.

Sprague, R.H. and Watson, H.J., 1983. Bit by bit: towards decision support systems. In: W.C. House (Editor), Decision Support Systems. Petrocelli, New York, pp. 15–32.

Wagner, H., 1983. Computergestützte Ist-Kosten-Leistungsrechnung für landwirtschaftliche Betriebe. Dissertation, Gießen, 273 pp.

Wagner, P. and Langenbruch, F., 1987. Die Anwendung eines Datenbanksystemes im landwirtschaftlichen Betrieb. Ber. Landwirtsch., 65: 140–168.